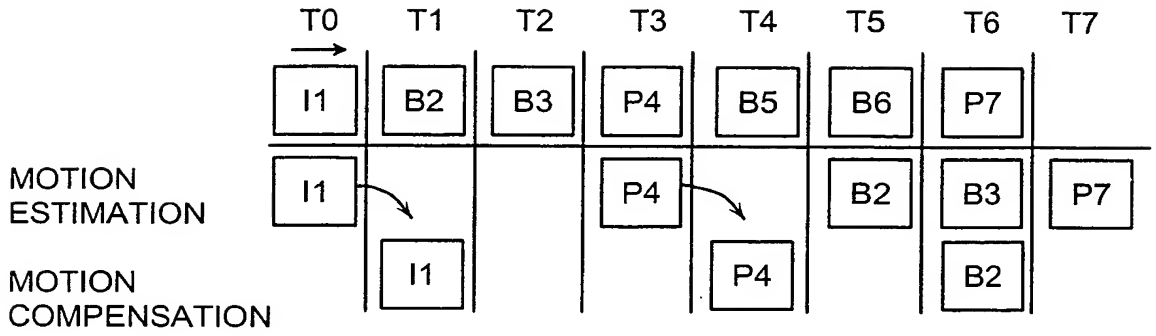


FIG. 1
PRIOR ART

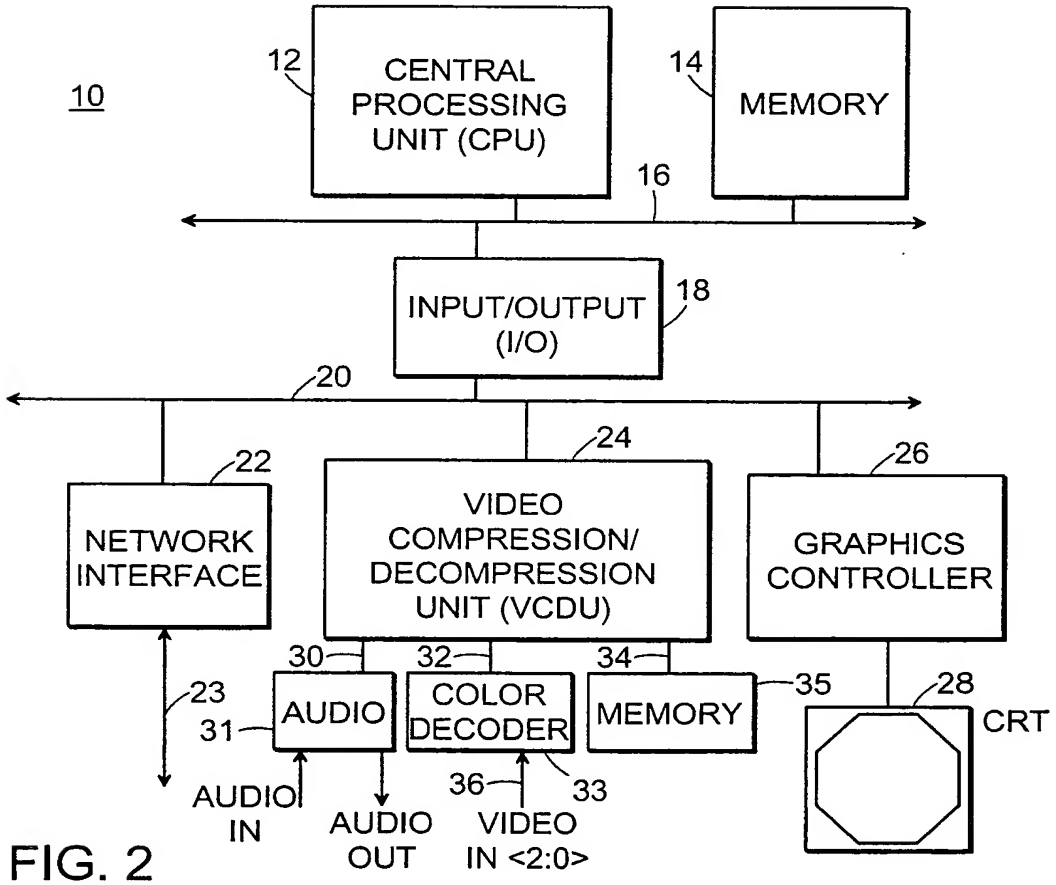


FIG. 2

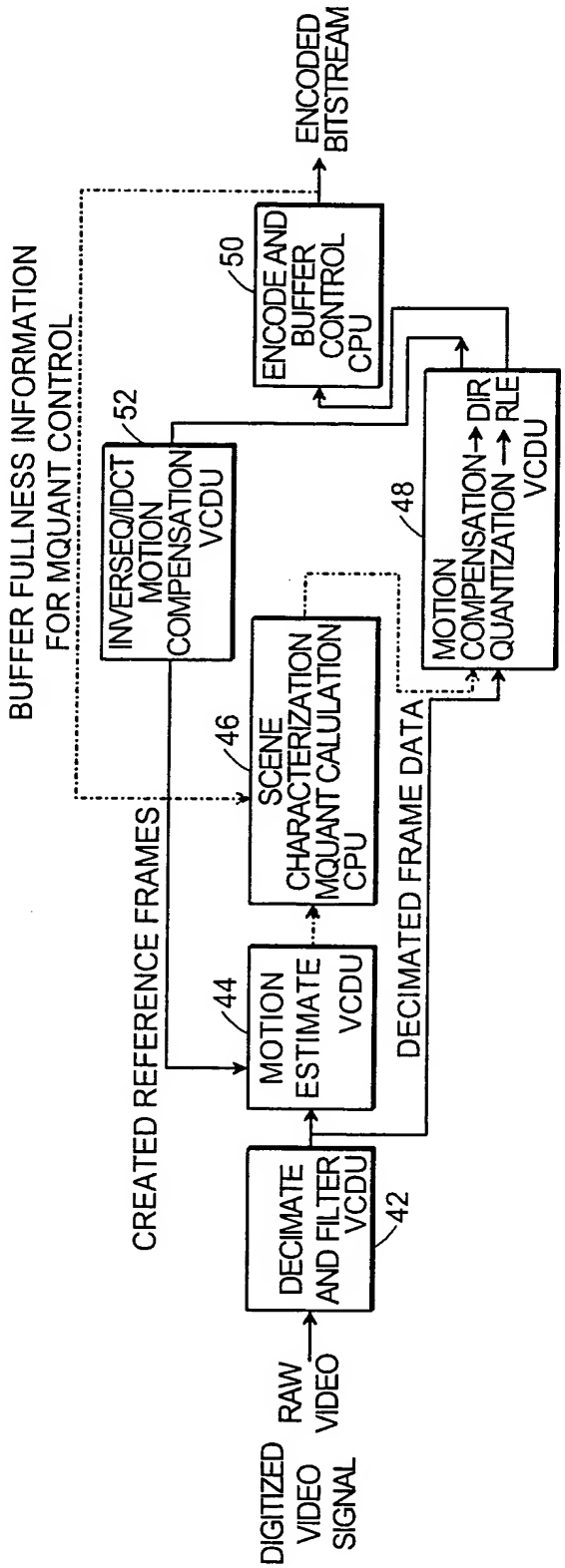

DIGITIZED VIDEO SIGNAL
RAW VIDEO
DECIMATE AND FILTER VCDU
42
MOTION ESTIMATE VCDU
44
SCENE CHARACTERIZATION MQUANT CALULATION CPU
46
INVERSEQ/IDCT MOTION COMPENSATION VCDU
52
ENCODE AND BUFFER CONTROL CPU
50
MOTION COMPENSATION→DIR QUANTIZATION→RLE VCDU
48
CREATED REFERENCE FRAMES
DECIMATED FRAME DATA
BUFFER FULLNESS INFORMATION FOR MQUANT CONTROL
ENCODED BITSTREAM
HIGH BANDWIDTH DATA (1 TO 4 MBYTES/SEC)
LOW BANDWIDTH DATA (<.5 MBYTES/SEC)

FIG. 3

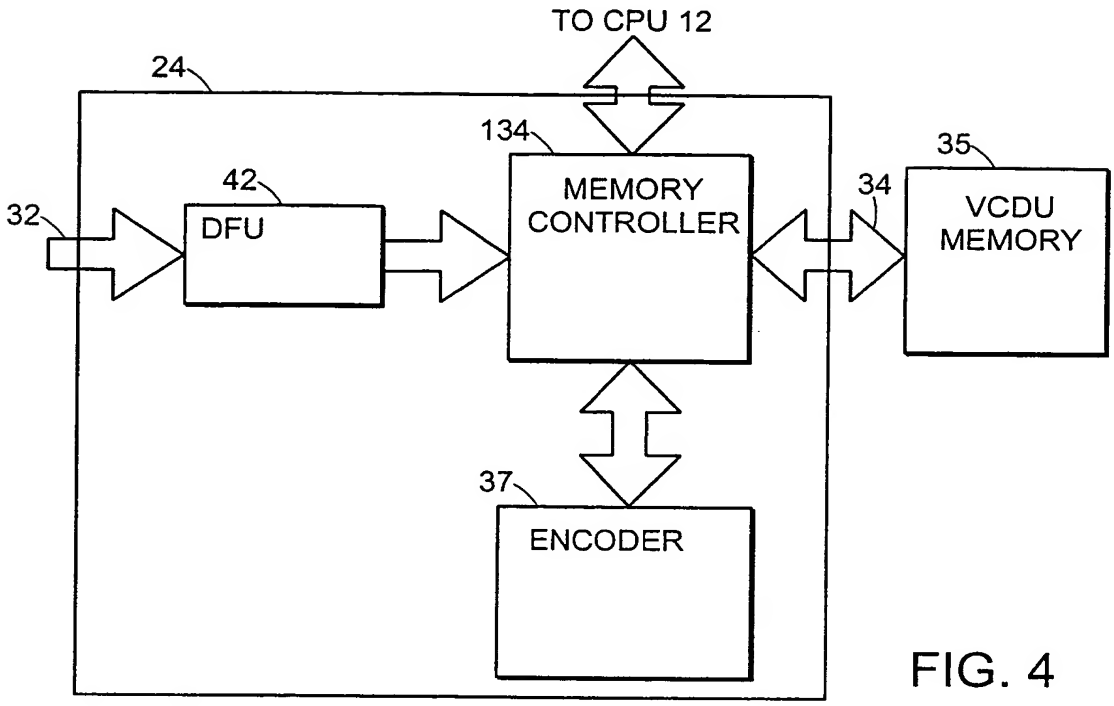

TO CPU 12
24
134
35
42
MEMORY
CONTROLLER
34
32
DFU
VCDU
MEMORY
37
ENCODER

FIG. 4

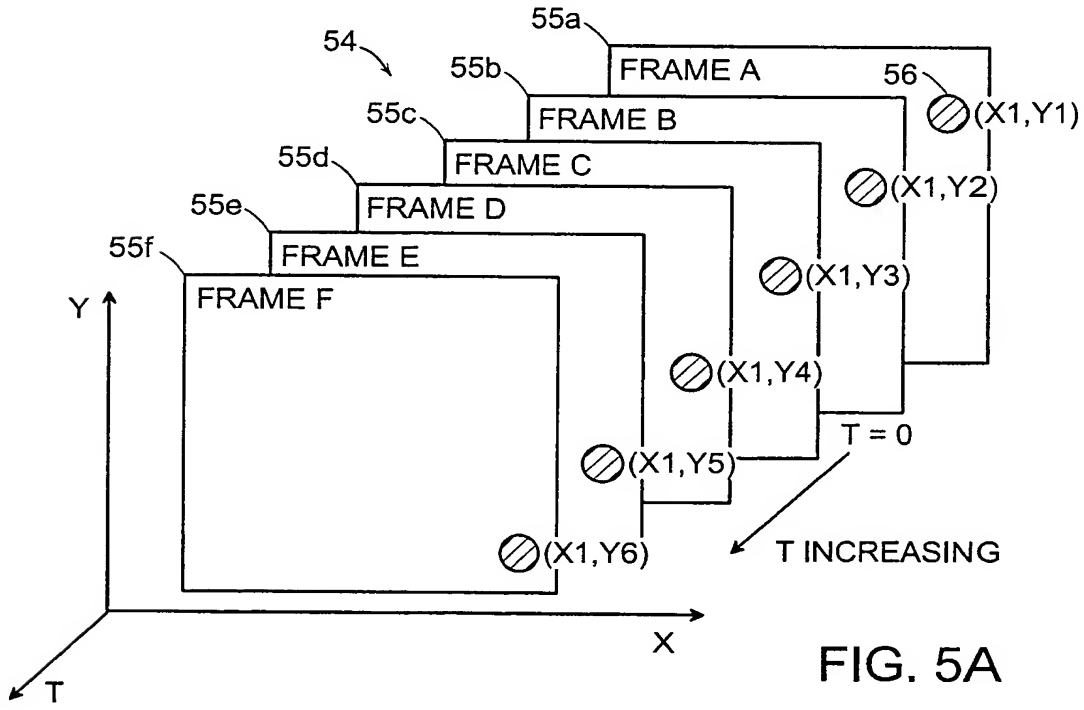

55a
54
55b
FRAME A
56
(X1,Y1)
55c
FRAME B
55d
FRAME C
(X1,Y2)
55e
FRAME D
55f
FRAME E
(X1,Y3)
Y
FRAME F
(X1,Y4)
T = 0
(X1,Y5)
(X1,Y6)
T INCREASING
X
T

FIG. 5A

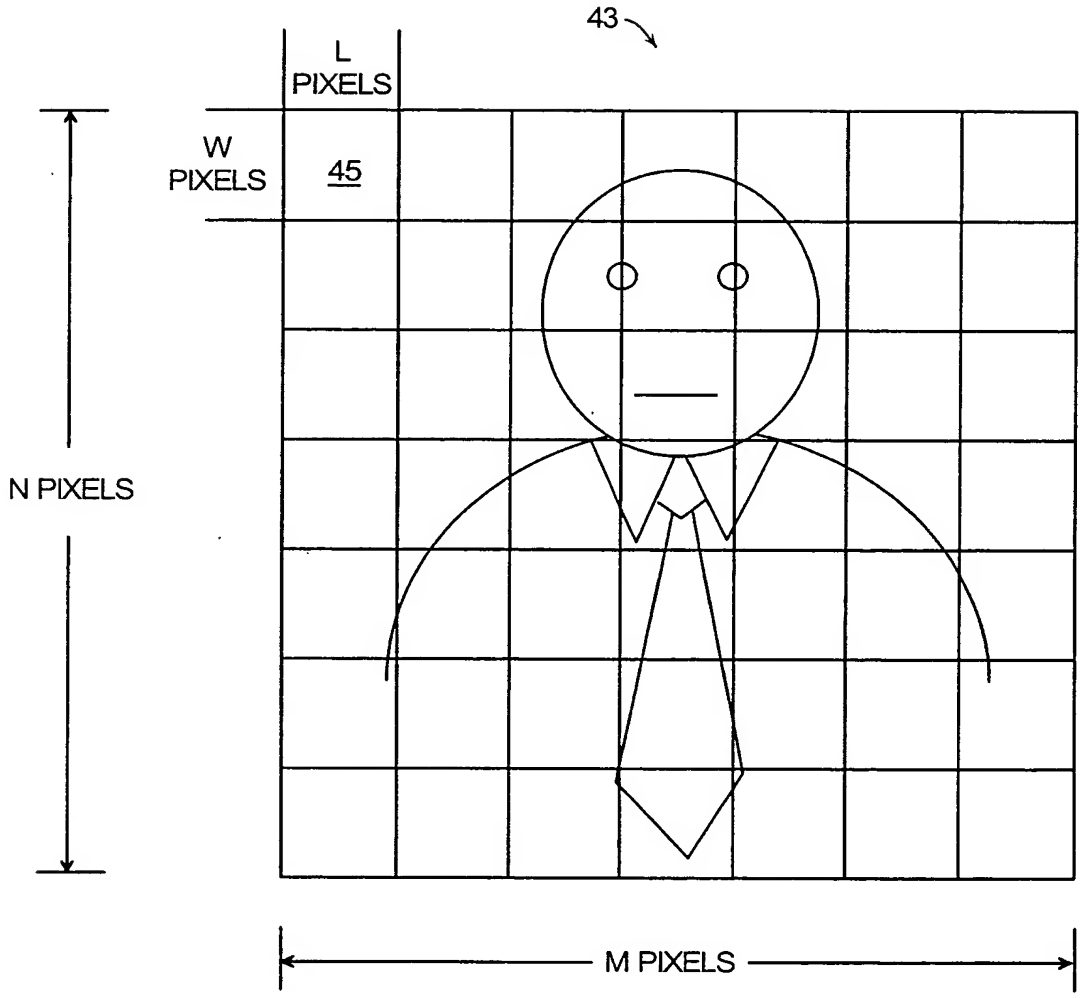


FIG. 5B

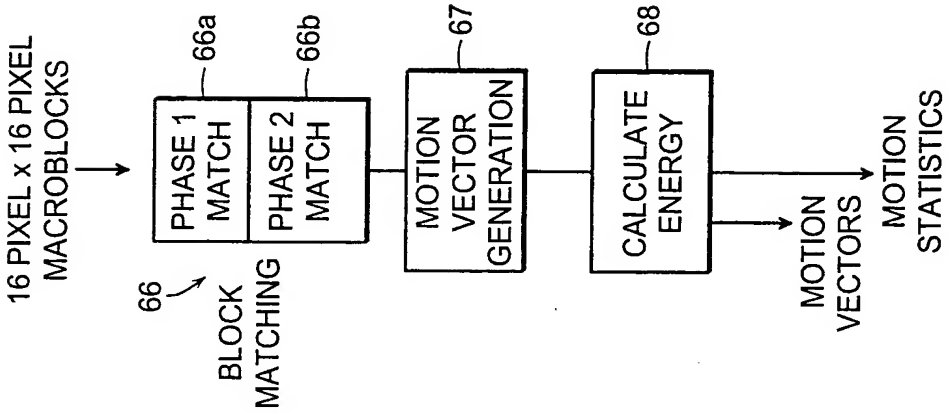

16 PIXEL x 16 PIXEL MACROBLOCKS
66
BLOCK MATCHING
PHASE 1 MATCH
66a
PHASE 2 MATCH
66b
MOTION VECTOR GENERATION
67
CALCULATE ENERGY
68
MOTION VECTORS
MOTION STATISTICS

FIG. 6B

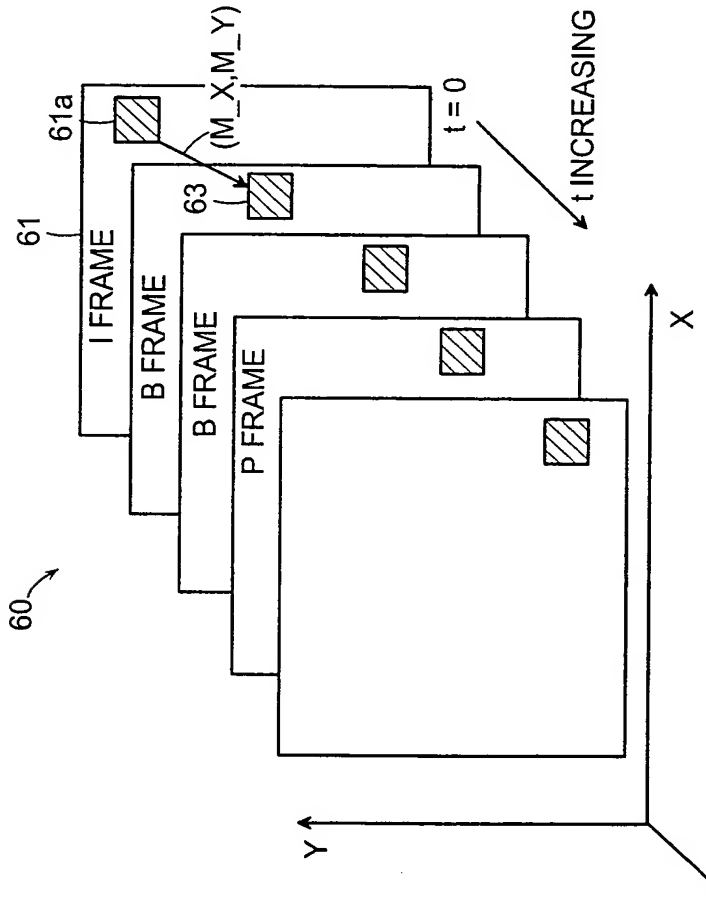

60
61
I FRAME
B FRAME
B FRAME
P FRAME
61a
63
(M_X,M_Y)
t = 0
t INCREASING
X
Y
t

FIG. 6A

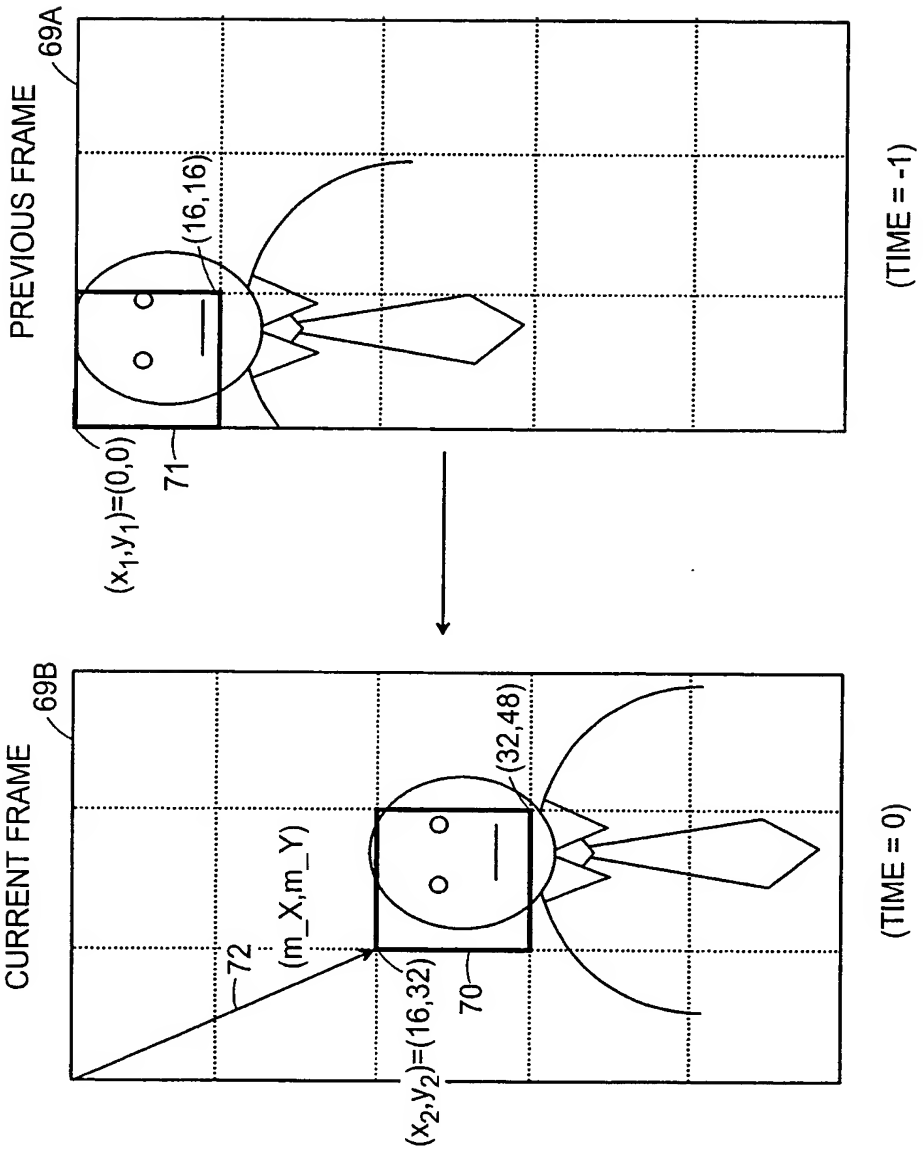

PREVIOUS FRAME
69A
(16,16)
(TIME = -1)
(x_1,y_1)=(0,0)
71
CURRENT FRAME
69B
(32,48)
(m_X,m_Y)
72
(TIME = 0)
70
(x_2,y_2)=(16,32)

FIG. 6C

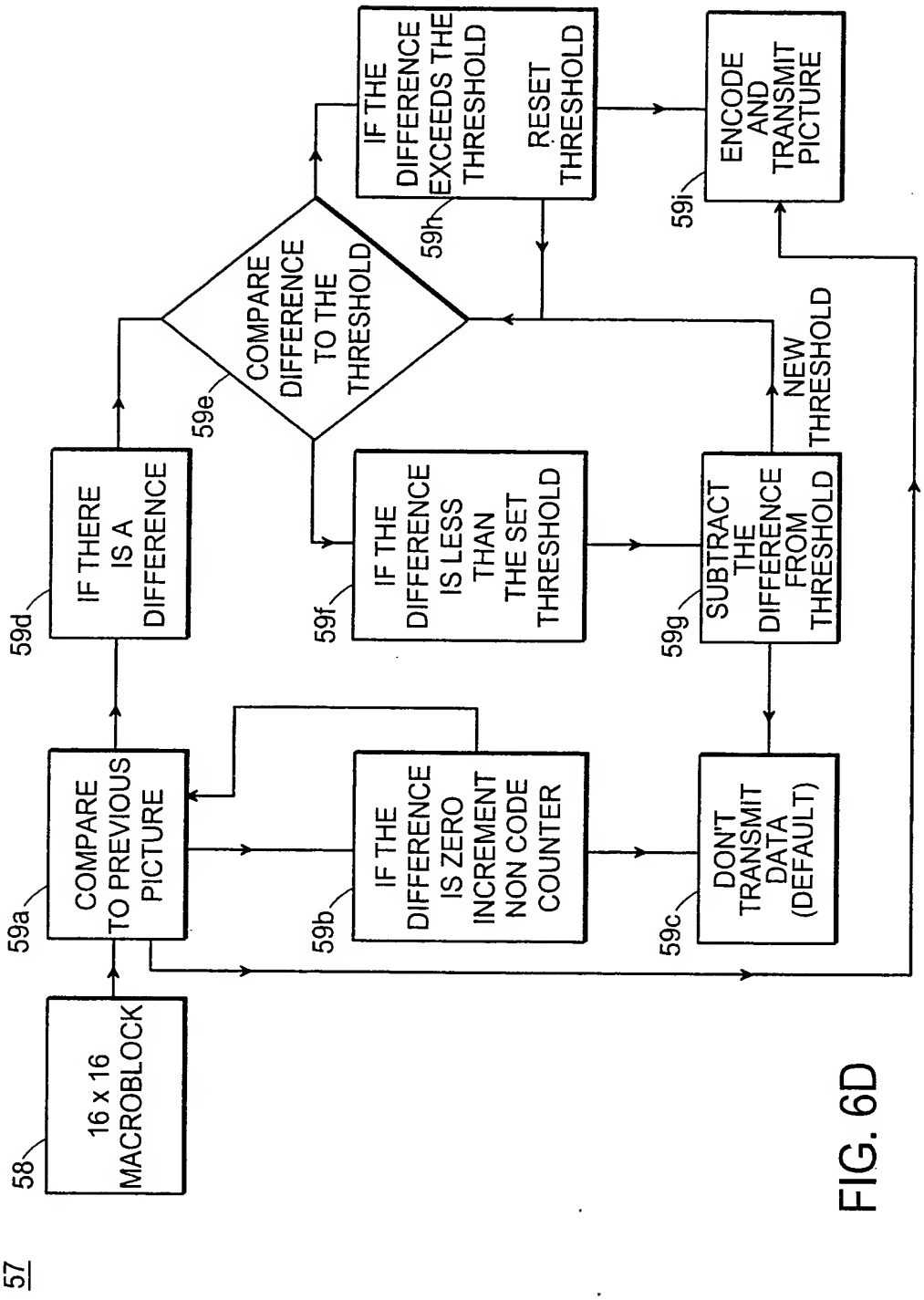

57
58
16 x 16 MACROBLOCK
59a
COMPARE TO PREVIOUS PICTURE
59b
IF THE DIFFERENCE IS ZERO INCREMENT NON CODE COUNTER
59c
DON'T TRANSMIT DATA (DEFAULT)
59d
IF THERE IS A DIFFERENCE
59e
COMPARE DIFFERENCE TO THE THRESHOLD
59f
IF THE DIFFERENCE IS LESS THAN THE SET THRESHOLD
59g
SUBTRACT THE DIFFERENCE FROM THRESHOLD
NEW THRESHOLD
59h
IF THE DIFFERENCE EXCEEDS THE THRESHOLD
RESET THRESHOLD
59i
ENCODE AND TRANSMIT PICTURE

FIG. 6D

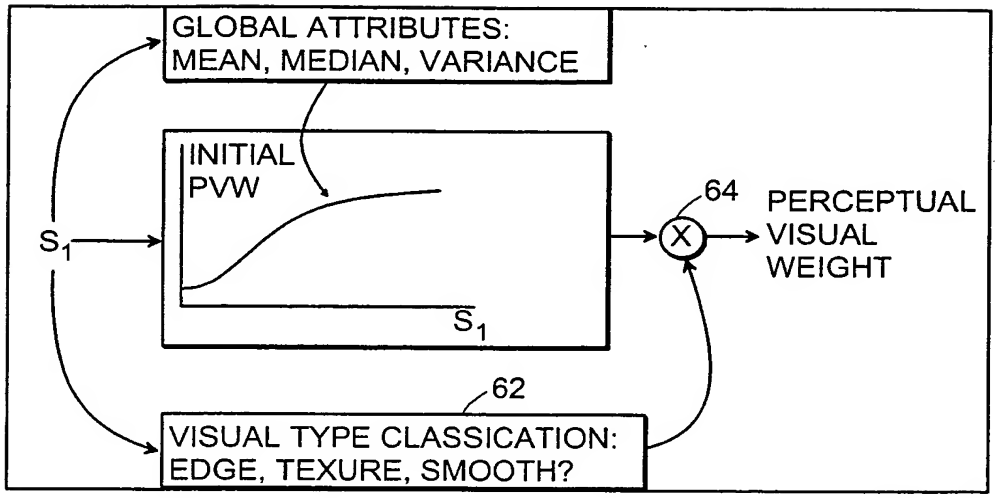

GLOBAL ATTRIBUTES:
MEAN, MEDIAN, VARIANCE
INITIAL
PVW
S_1
S_1
64
X
PERCEPTUAL
VISUAL
WEIGHT
62
VISUAL TYPE CLASSICATION:
EDGE, TEXURE, SMOOTH?

FIG. 7

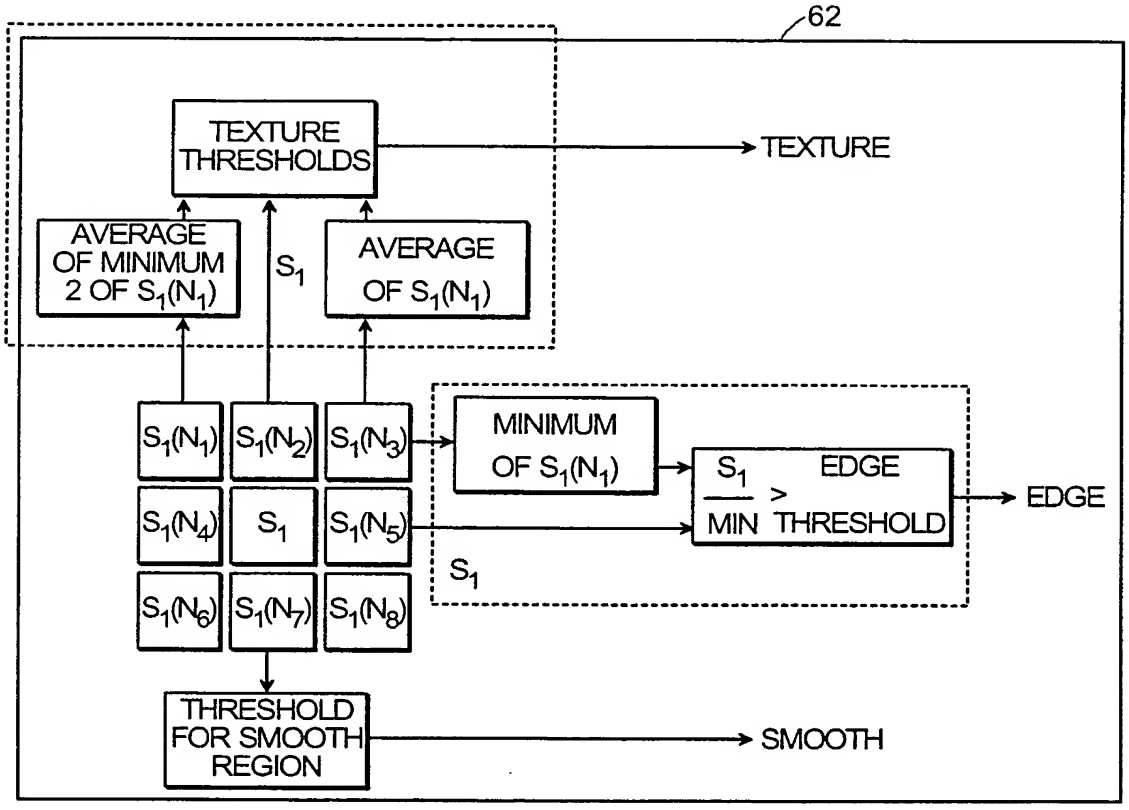

62
TEXTURE
THRESHOLDS
TEXTURE
AVERAGE
OF MINIMUM
2 OF $S_1(N_1)$
S_1
AVERAGE
OF $S_1(N_1)$
$S_1(N_1)$
$S_1(N_2)$
$S_1(N_3)$
$S_1(N_4)$
S_1
$S_1(N_5)$
$S_1(N_6)$
$S_1(N_7)$
$S_1(N_8)$
MINIMUM
OF $S_1(N_1)$
S_1
EDGE
MIN THRESHOLD
EDGE
S_1
THRESHOLD
FOR SMOOTH
REGION
SMOOTH

FIG. 8

PROCESS FOR INITIAL PVW DETERMINATION FOR I-CODED FRAMES

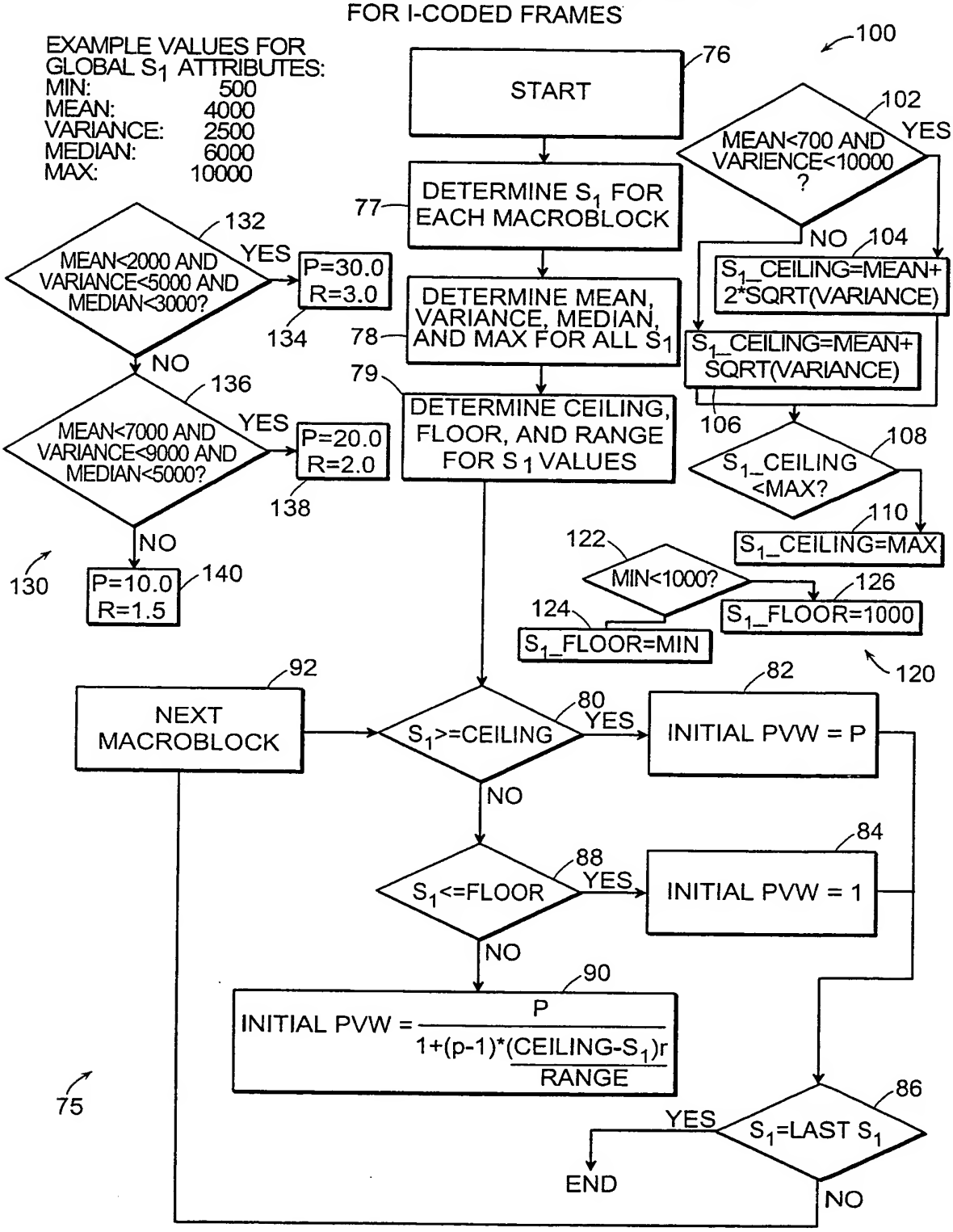


FIG. 9

PROCESS FOR CLASSIFYING INTRACODED FRAMES OR PICTURES

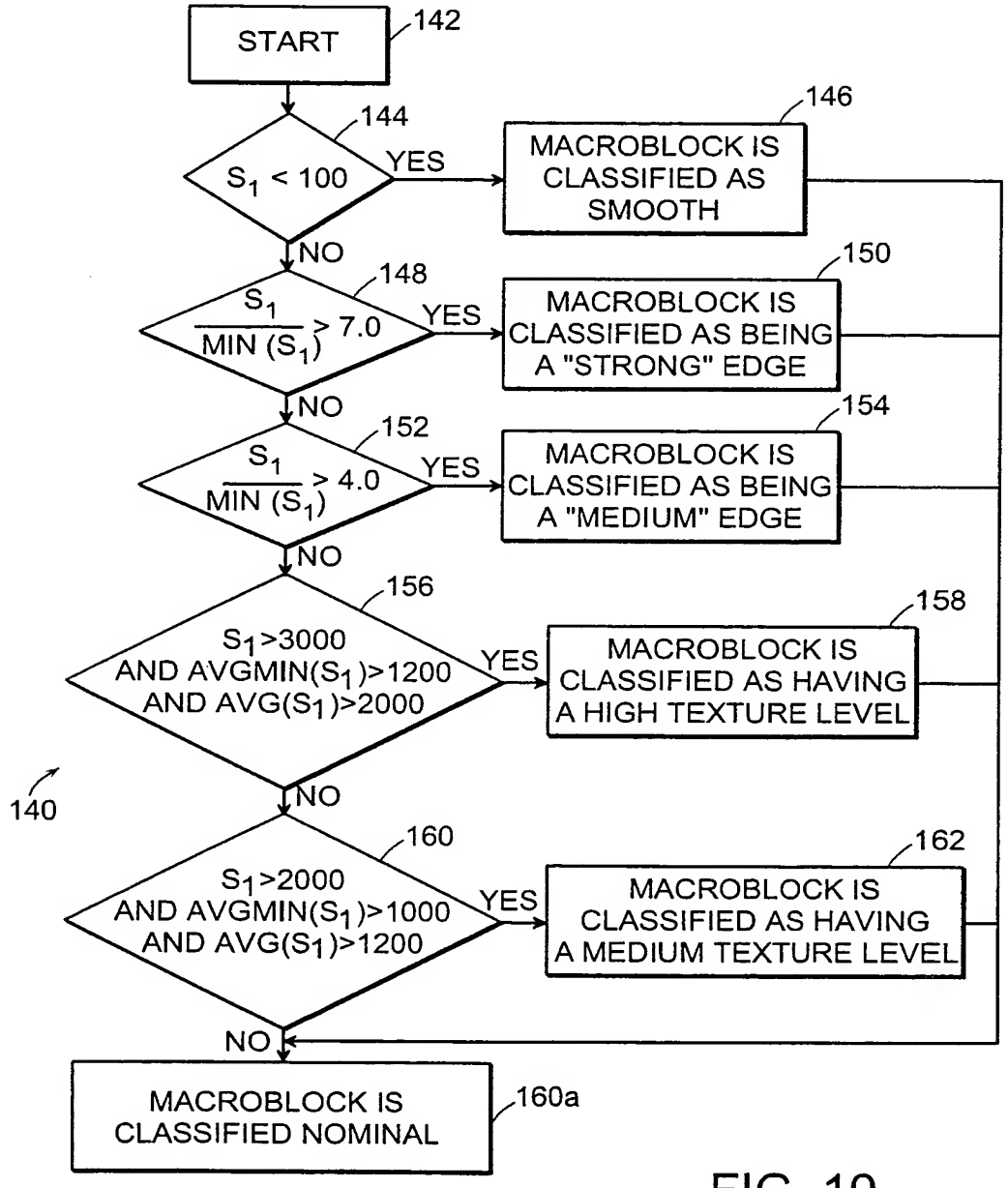


FIG. 10

PROCESS FOR PROVIDING FINAL PVW VALUE FOR I FRAMES

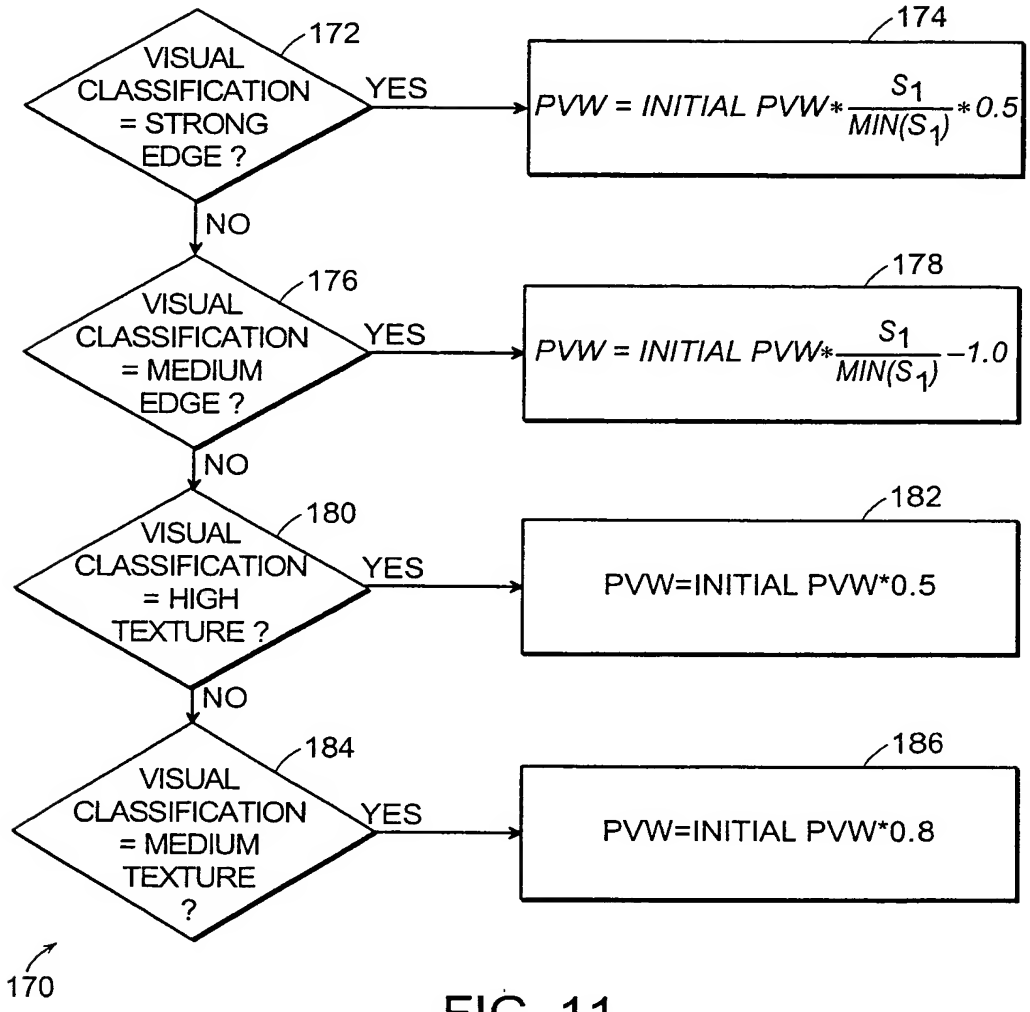


FIG. 11

PROCESS FOR INITIAL PVW DETERMINATION
FOR P-CODED FRAMES

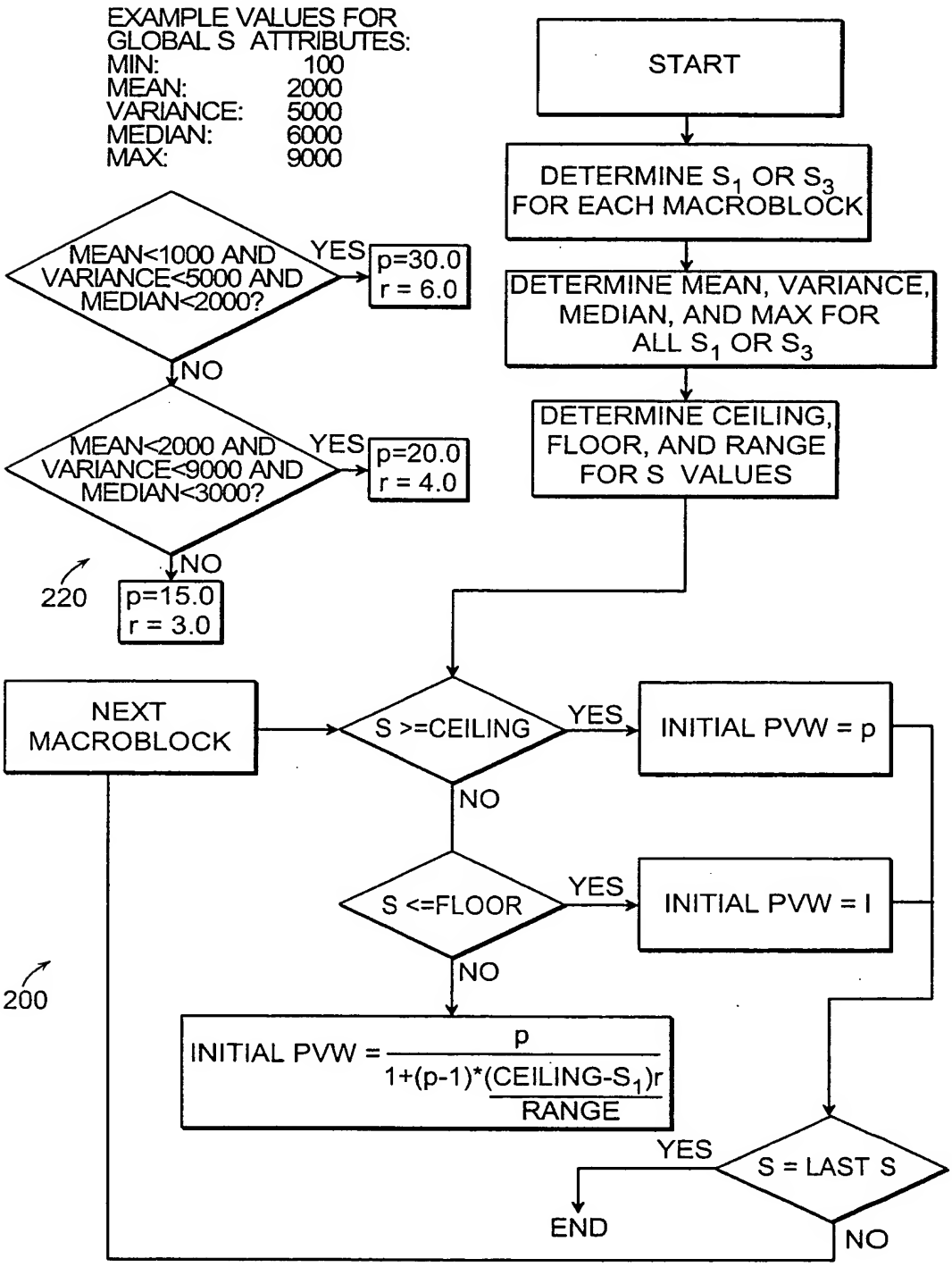


FIG. 12

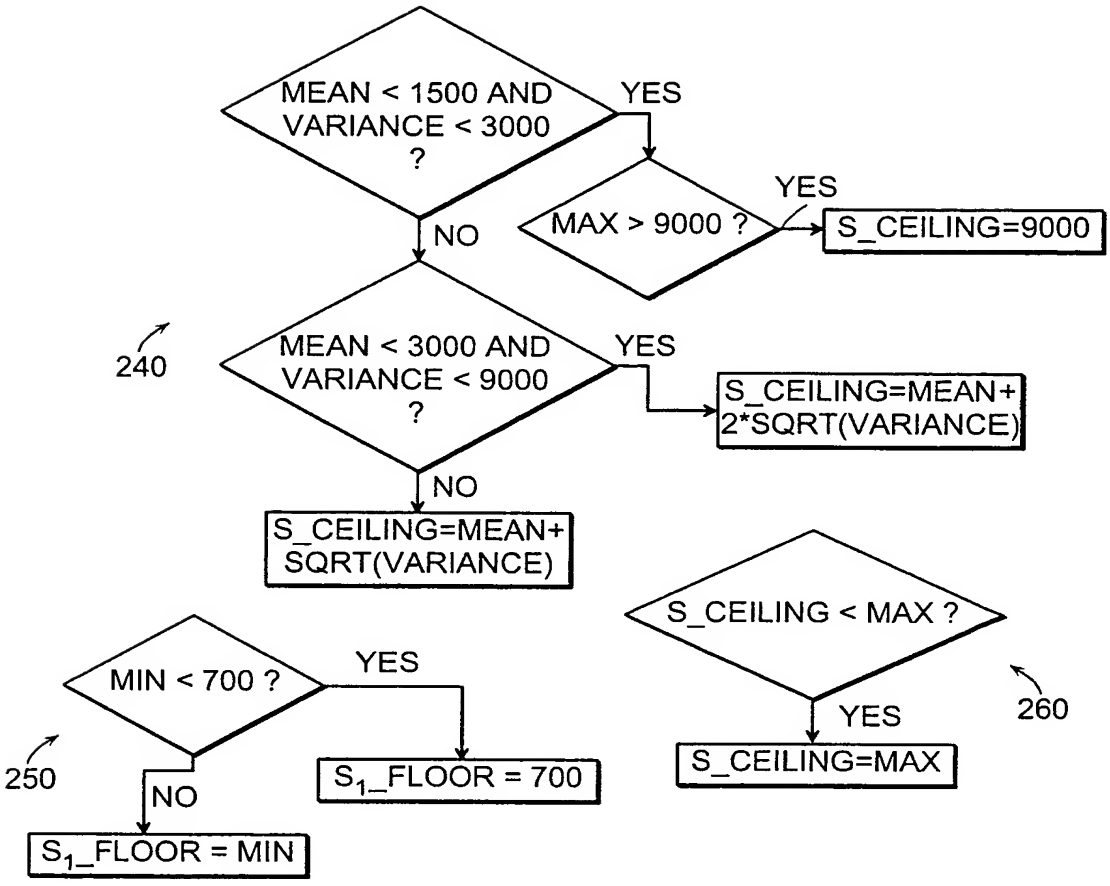

MEAN < 1500 AND VARIANCE < 3000 ?
YES
MAX > 9000 ?
YES
S_CEILING=9000
NO
240
MEAN < 3000 AND VARIANCE < 9000 ?
YES
S_CEILING=MEAN+ 2*SQRT(VARIANCE)
NO
S_CEILING=MEAN+ SQRT(VARIANCE)
S_CEILING < MAX ?
260
YES
S_CEILING=MAX
MIN < 700 ?
YES
250
NO
S1_FLOOR = 700
S1_FLOOR = MIN

FIG. 13

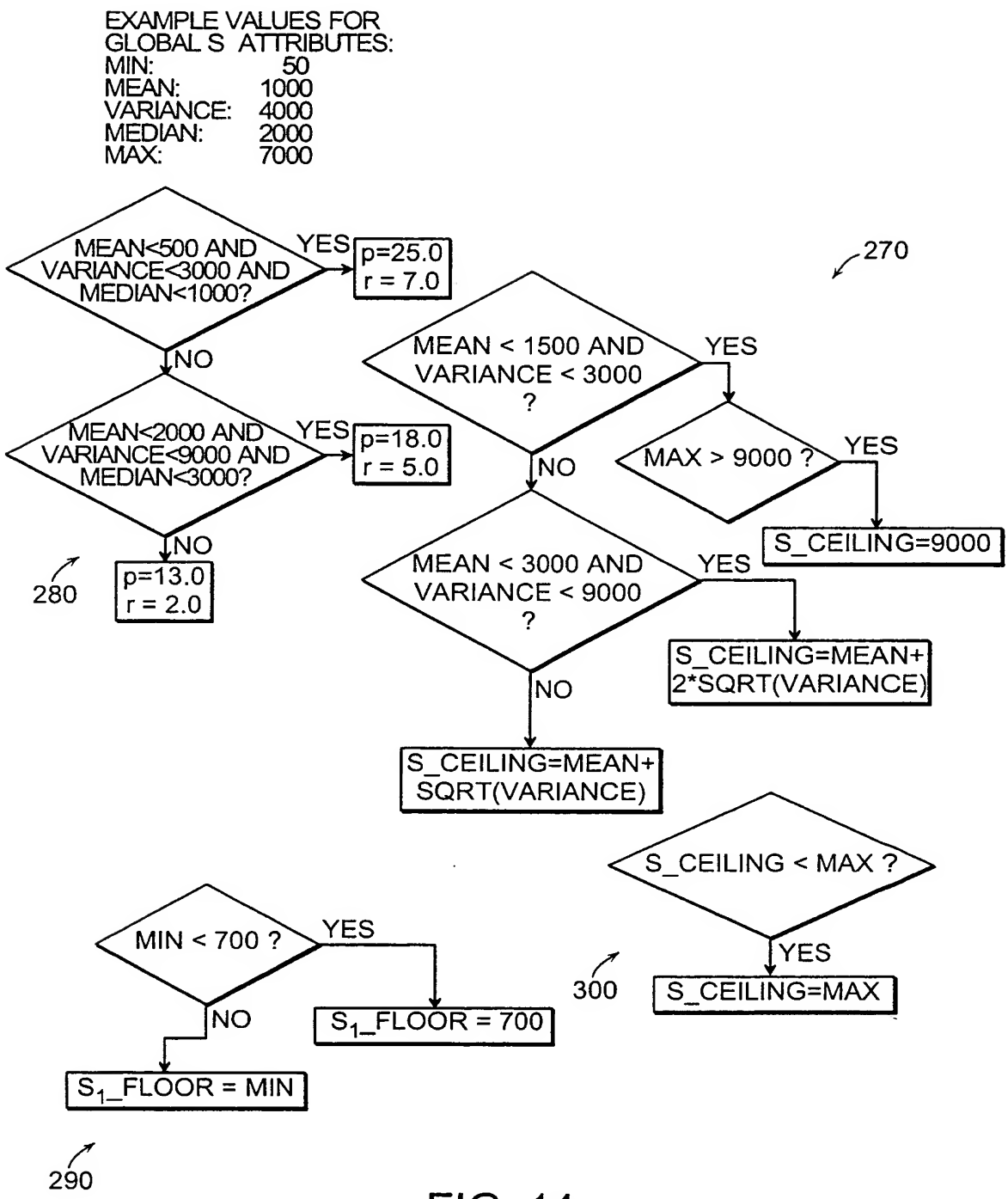

EXAMPLE VALUES FOR
GLOBAL S ATTRIBUTES:
MIN: 50
MEAN: 1000
VARIANCE: 4000
MEDIAN: 2000
MAX: 7000
MEAN<500 AND VARIANCE<3000 AND MEDIAN<1000?
YES
p=25.0
r = 7.0
NO
MEAN<2000 AND VARIANCE<9000 AND MEDIAN<3000?
YES
p=18.0
r = 5.0
NO
p=13.0
r = 2.0
280
270
MEAN < 1500 AND VARIANCE < 3000 ?
YES
NO
MAX > 9000 ?
YES
S_CEILING=9000
MEAN < 3000 AND VARIANCE < 9000 ?
YES
S_CEILING=MEAN+ 2*SQRT(VARIANCE)
NO
S_CEILING=MEAN+ SQRT(VARIANCE)
S_CEILING < MAX ?
YES
S_CEILING=MAX
300
MIN < 700 ?
YES
S1_FLOOR = 700
NO
S1_FLOOR = MIN
290

FIG. 14

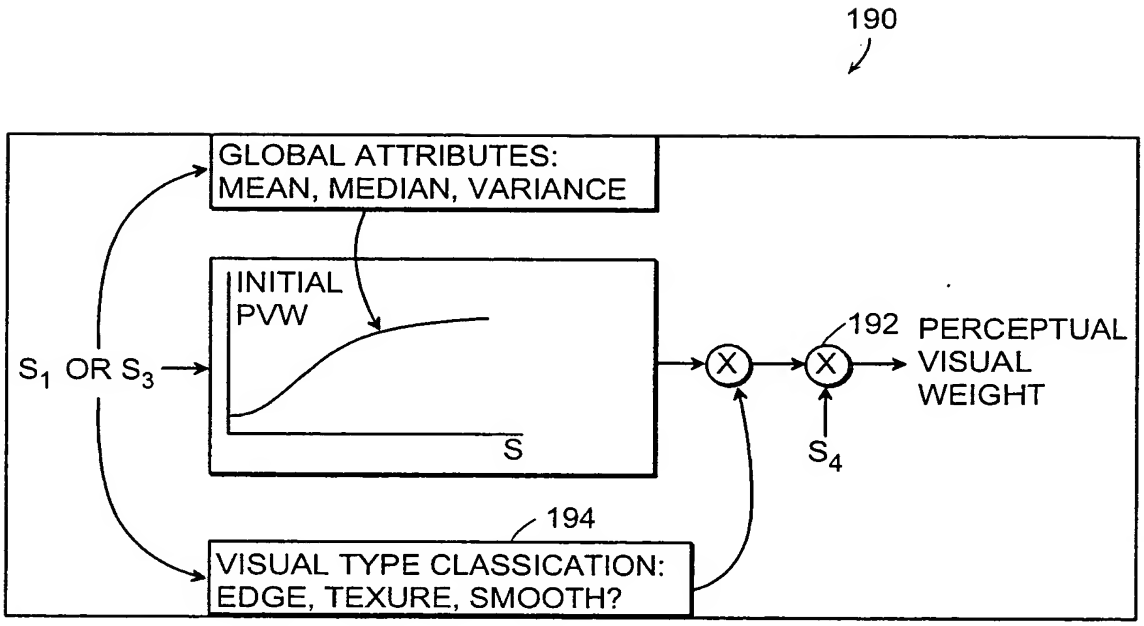

190
GLOBAL ATTRIBUTES:
MEAN, MEDIAN, VARIANCE
INITIAL
PVW
S
S1 OR S3
X
X
192
PERCEPTUAL
VISUAL
WEIGHT
S4
194
VISUAL TYPE CLASSICATION:
EDGE, TEXURE, SMOOTH?

FIG. 15

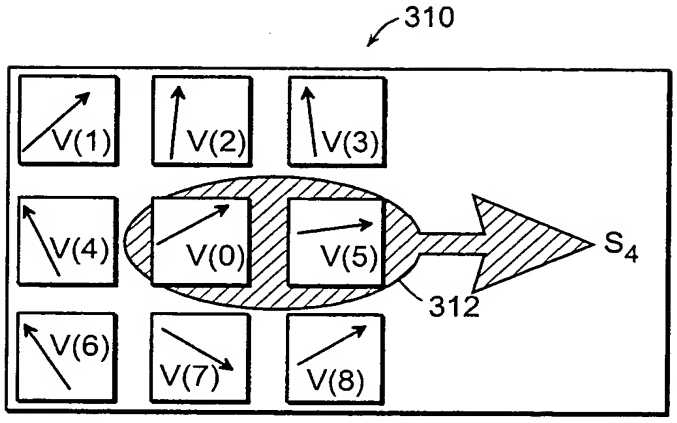

310
V(1)
V(2)
V(3)
V(4)
V(0)
V(5)
S_4
312
V(6)
V(7)
V(8)

FIG. 16

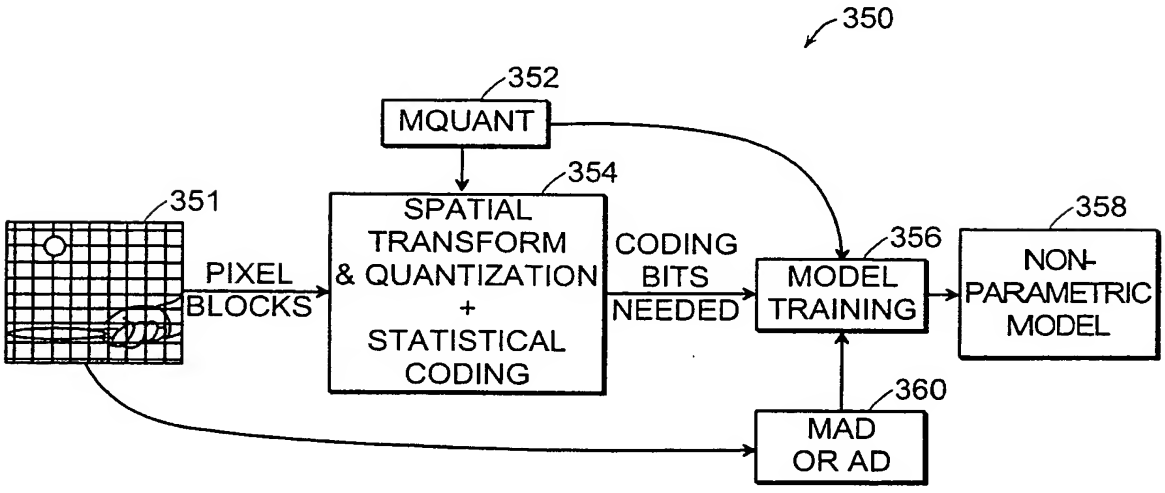

350
352
MQUANT
354
351
SPATIAL
TRANSFORM
& QUANTIZATION
+
STATISTICAL
CODING
PIXEL
BLOCKS
CODING
BITS
NEEDED
356
MODEL
TRAINING
358
NON-
PARAMETRIC
MODEL
360
MAD
OR AD

FIG. 17

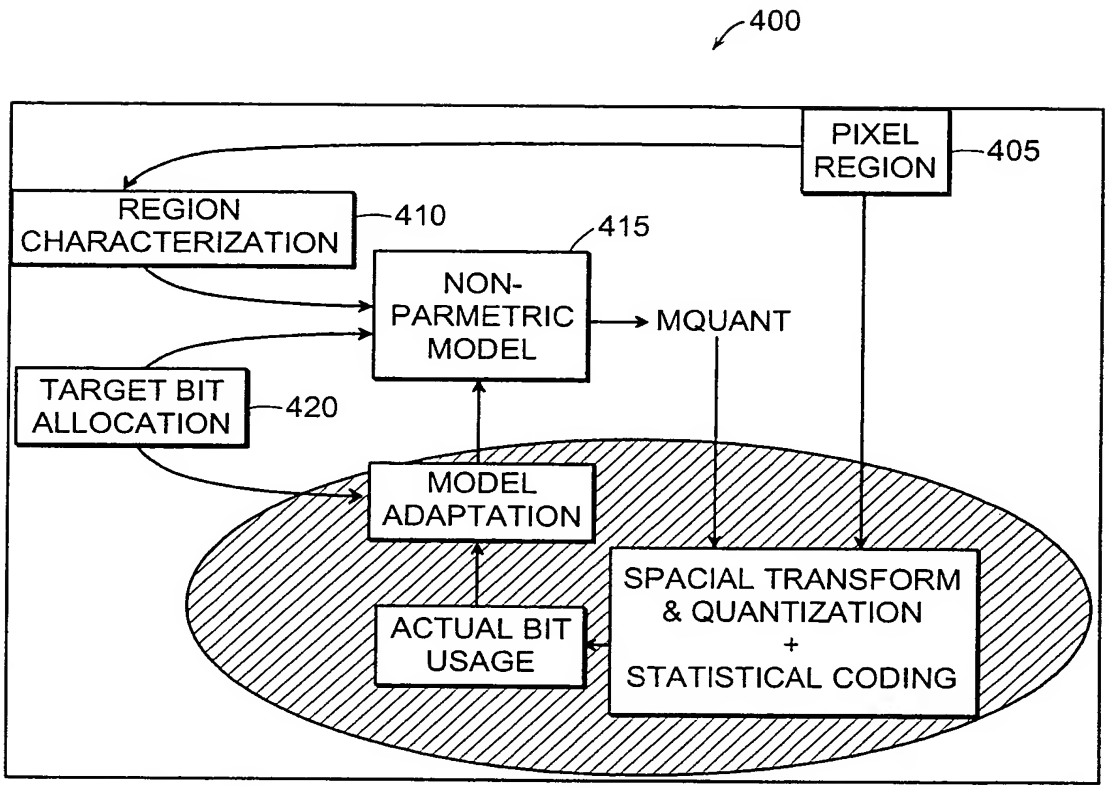
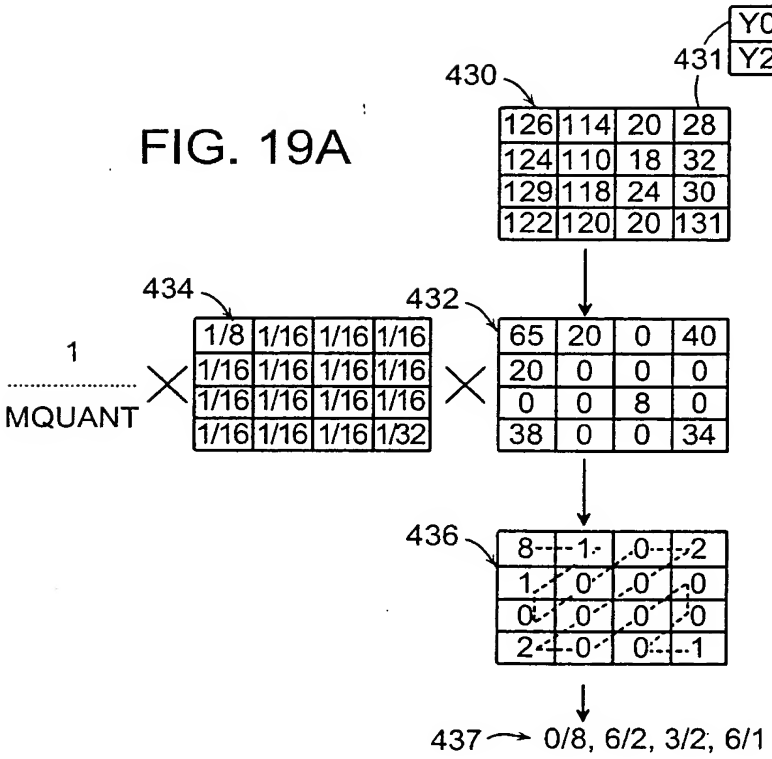

400
PIXEL REGION
405
REGION CHARACTERIZATION
410
415
NON-PARMETRIC MODEL
MQUANT
TARGET BIT ALLOCATION
420
MODEL ADAPTATION
SPACIAL TRANSFORM & QUANTIZATION + STATISTICAL CODING
ACTUAL BIT USAGE

FIG. 18

FIG. 19A



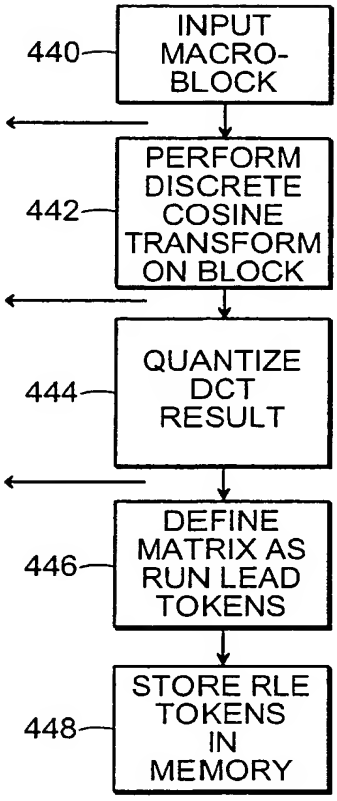


FIG. 19B

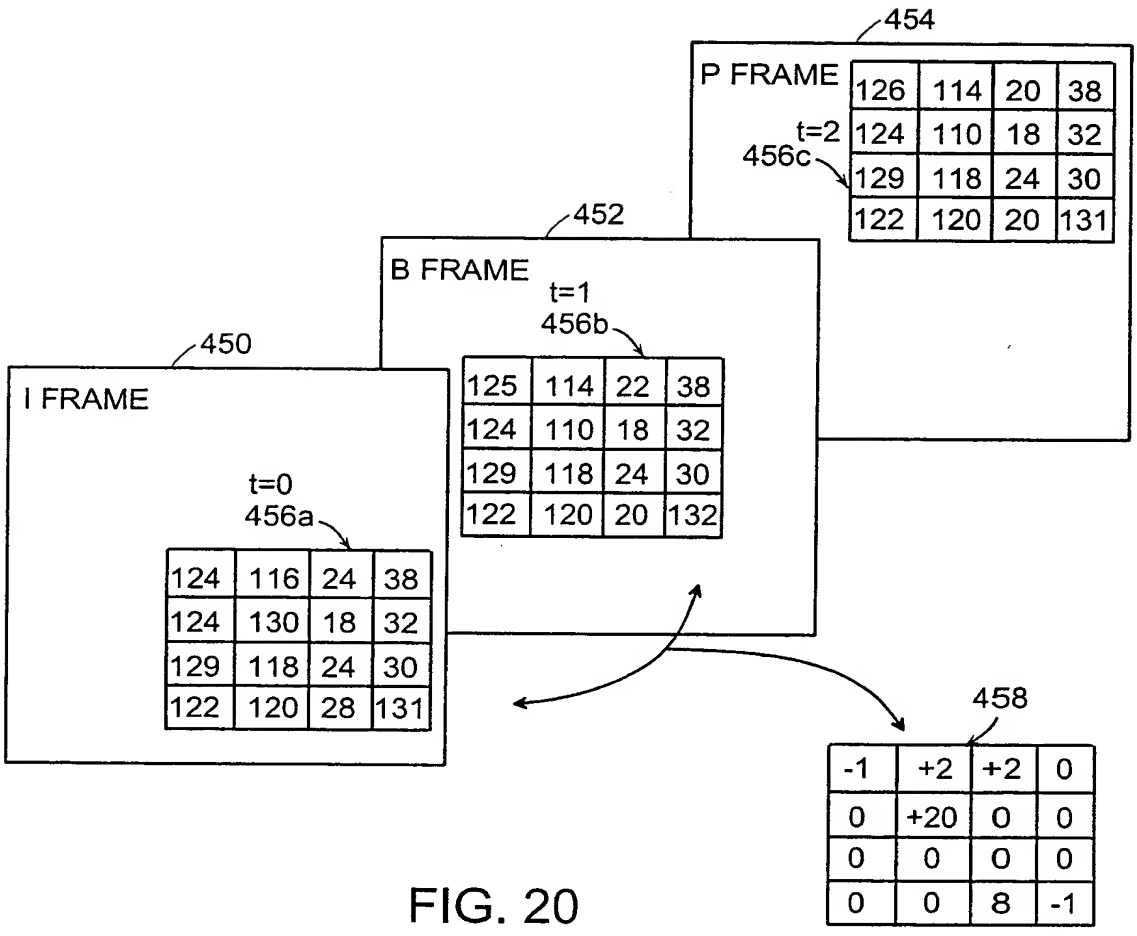

454
P FRAME
126 114 20 38
t=2
456c
124 110 18 32
129 118 24 30
122 120 20 131
452
B FRAME
t=1
456b
125 114 22 38
124 110 18 32
129 118 24 30
122 120 20 132
450
I FRAME
t=0
456a
124 116 24 38
124 130 18 32
129 118 24 30
122 120 28 131
458
-1 +2 +2 0
0 +20 0 0
0 0 0 0
0 0 8 -1

FIG. 20

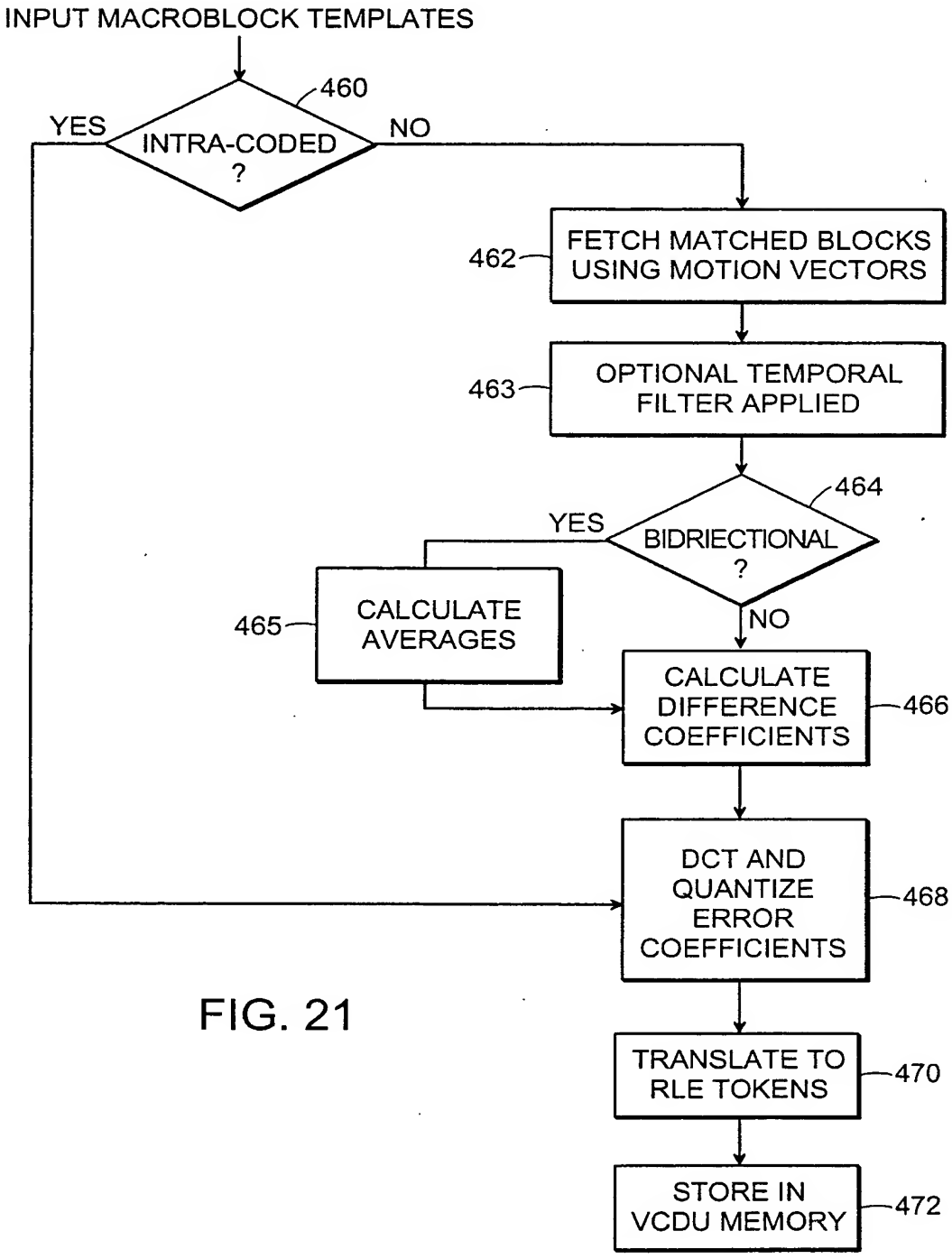

INPUT MACROBLOCK TEMPLATES
460
INTRA-CODED ?
YES
NO
462
FETCH MATCHED BLOCKS USING MOTION VECTORS
463
OPTIONAL TEMPORAL FILTER APPLIED
464
BIDRIECTIONAL ?
YES
NO
465
CALCULATE AVERAGES
CALCULATE DIFFERENCE COEFFICIENTS
466
DCT AND QUANTIZE ERROR COEFFICIENTS
468
TRANSLATE TO RLE TOKENS
470
STORE IN VCDU MEMORY
472

FIG. 21

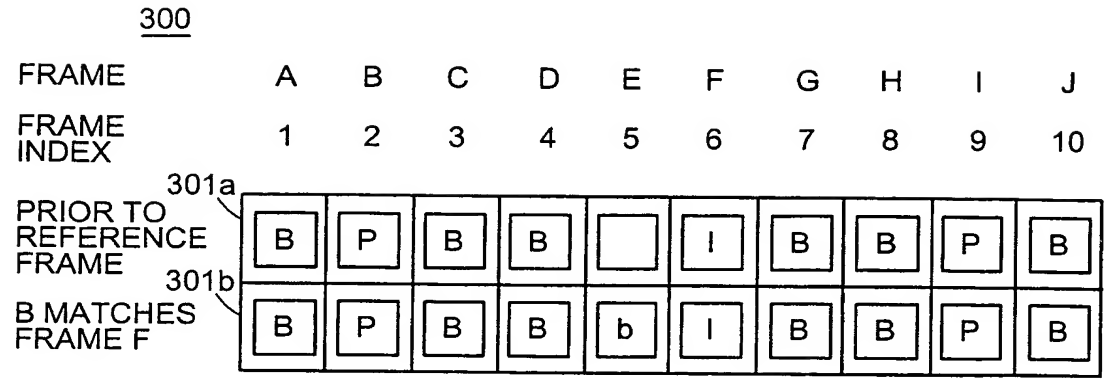


FIG. 22A

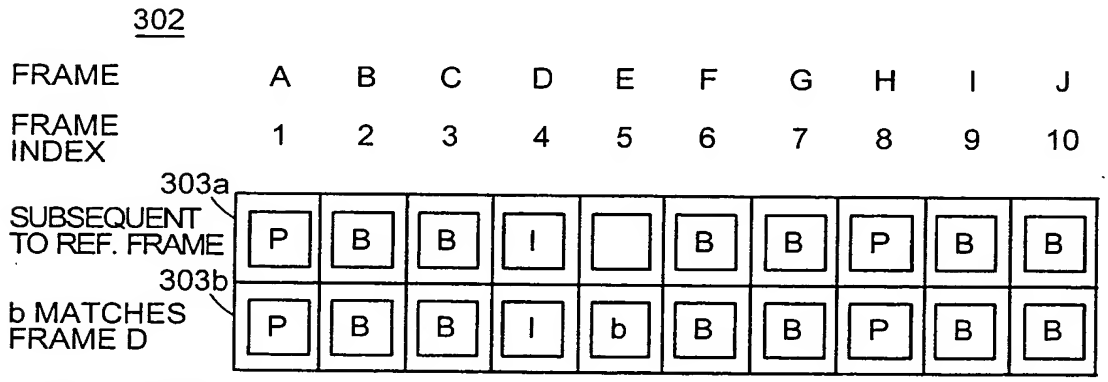


FIG. 22B

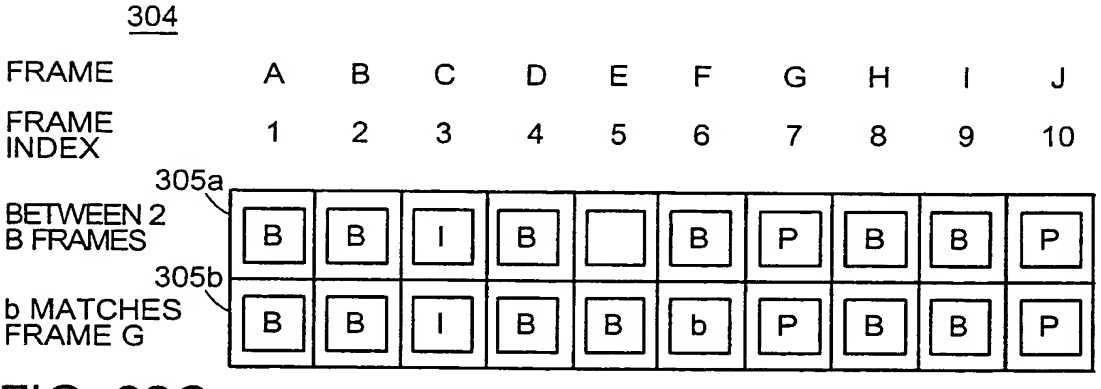


FIG. 22C

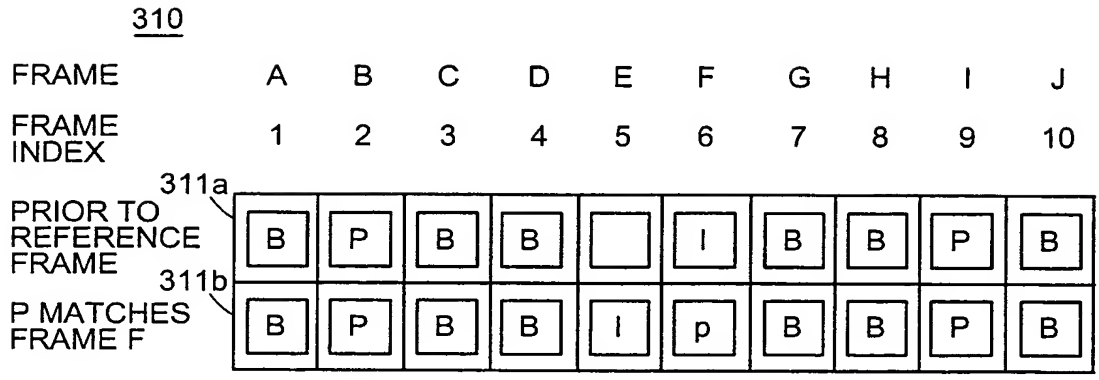


FIG. 23A

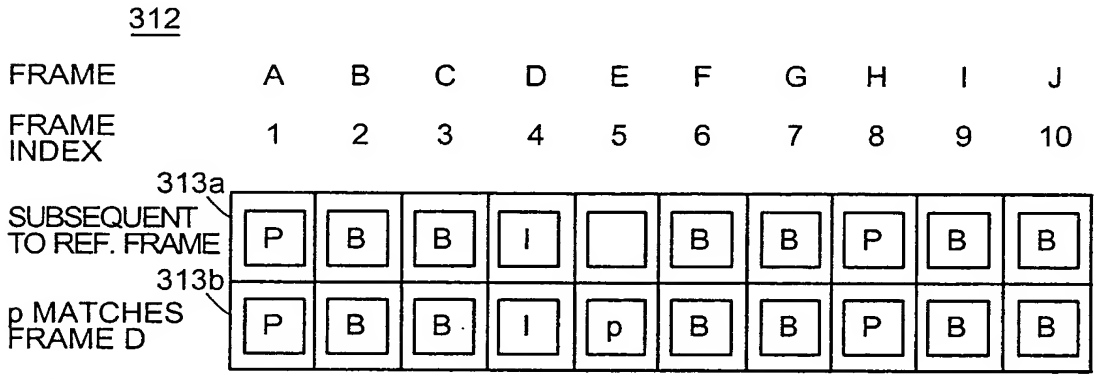


FIG. 23B

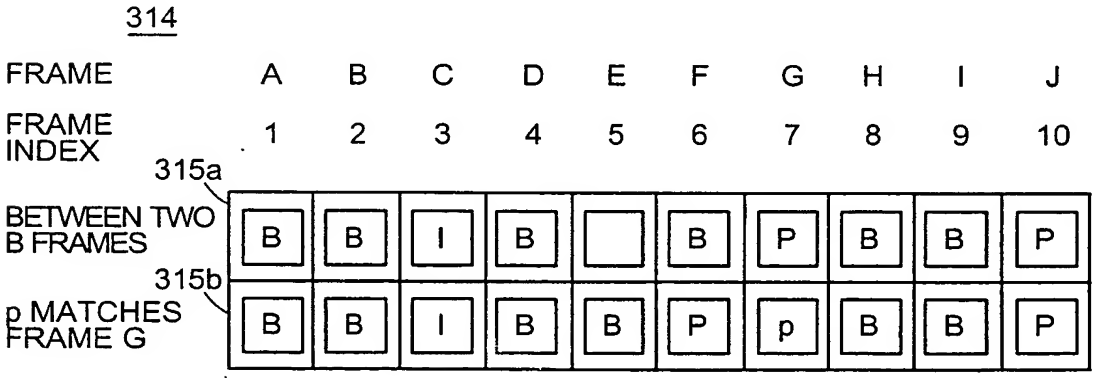


FIG. 23C

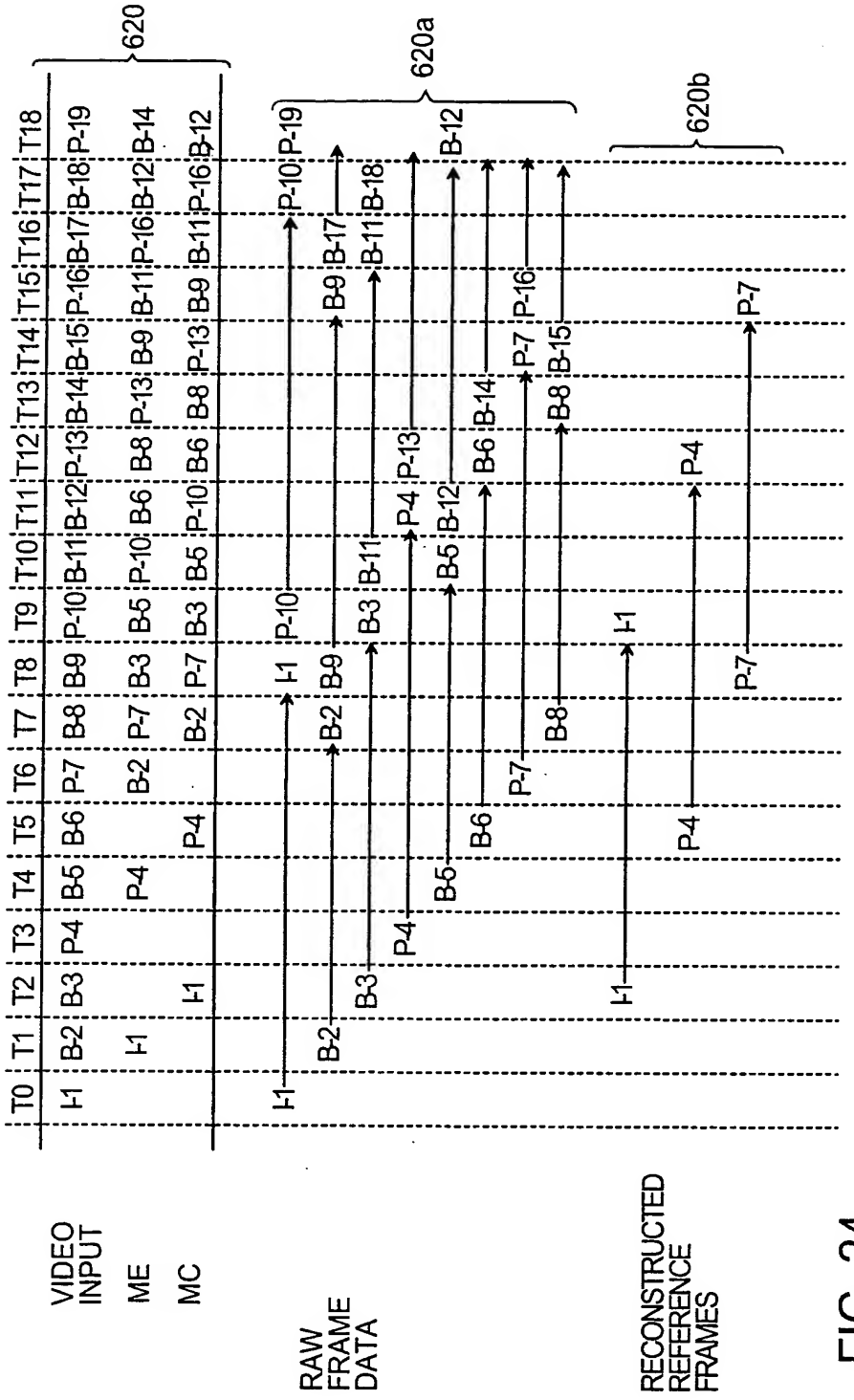

620
620a
620b
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18
I-1 B-2 B-3 P-4 B-5 B-6 P-7 B-8 B-9 P-10 B-11 B-12 P-13 B-14 B-15 P-16 B-17 B-18 P-19
I-1 B-2 P-4 B-5 P-7 B-8 B-9 P-10 B-11 B-12 P-13 B-14 B-15 P-16 B-17 B-18 P-19
I-1 B-2 B-3 P-4 B-5 B-6 P-7 B-8 B-9 P-10 B-11 B-12 P-13 B-14 B-15 P-16
VIDEO INPUT
ME
MC
RAW FRAME DATA
RECONSTRUCTED REFERENCE FRAMES

FIG. 24

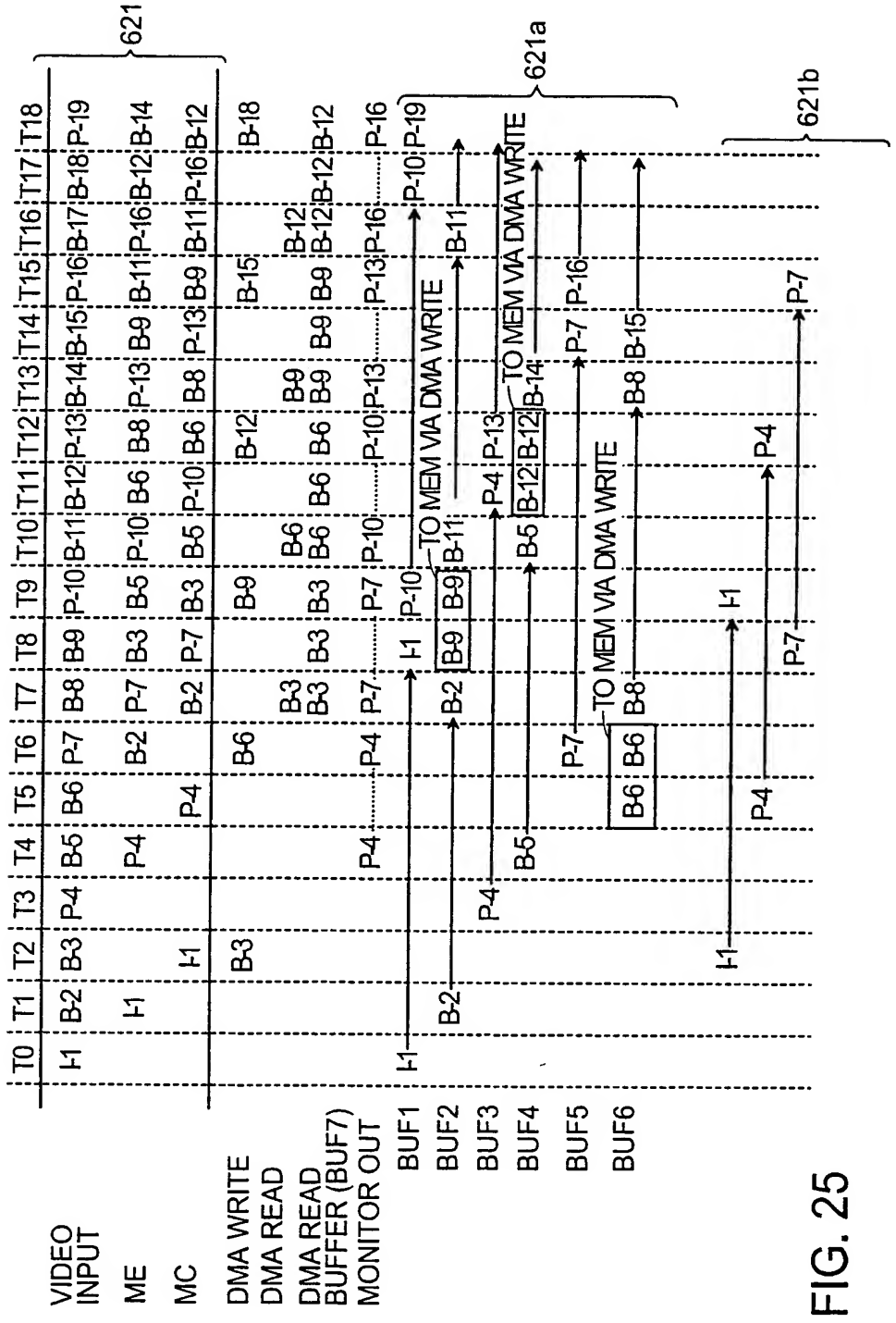

621
621a
621b
VIDEO INPUT
ME
MC
DMA WRITE
DMA READ
DMA READ BUFFER (BUF7)
MONITOR OUT
BUF1
BUF2
BUF3
BUF4
BUF5
BUF6
TO MEM VIA DMA WRITE
TO MEM VIA DMA WRITE
TO MEM VIA DMA WRITE
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18

FIG. 25

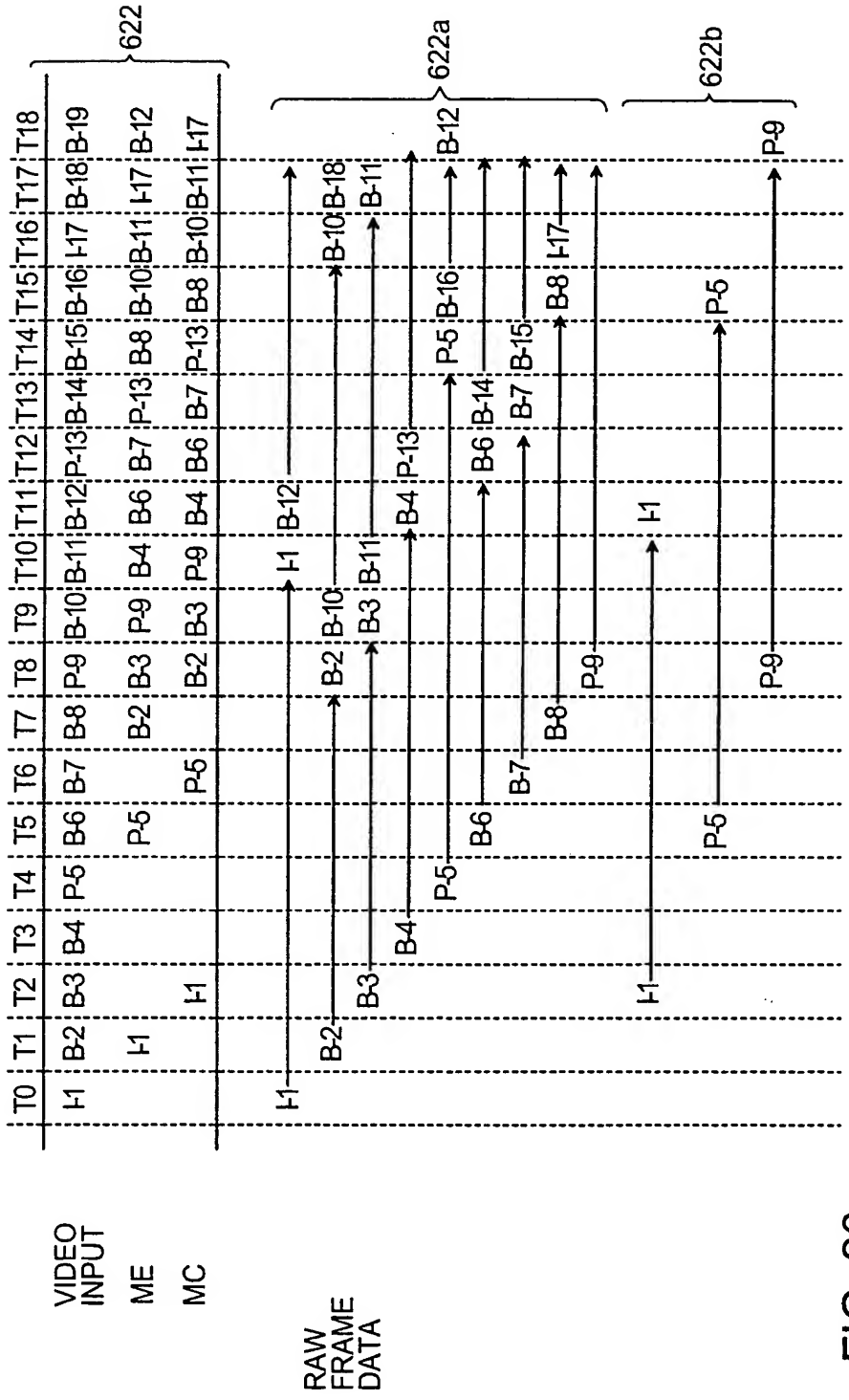

622
622a
622b
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18
I-1 B-2 B-3 B-4 P-5 B-6 B-7 B-8 P-9 B-10 B-11 B-12 P-13 B-14 B-15 B-16 I-17 B-18 B-19
I-1 P-5 B-2 B-3 P-9 B-4 B-6 P-13 B-8 B-10 B-11 B-12
I-1 P-5 B-3 B-4 P-9 B-6 B-7 P-13 B-8 B-10 B-11 I-17
VIDEO INPUT
ME
MC
RAW FRAME DATA

FIG. 26

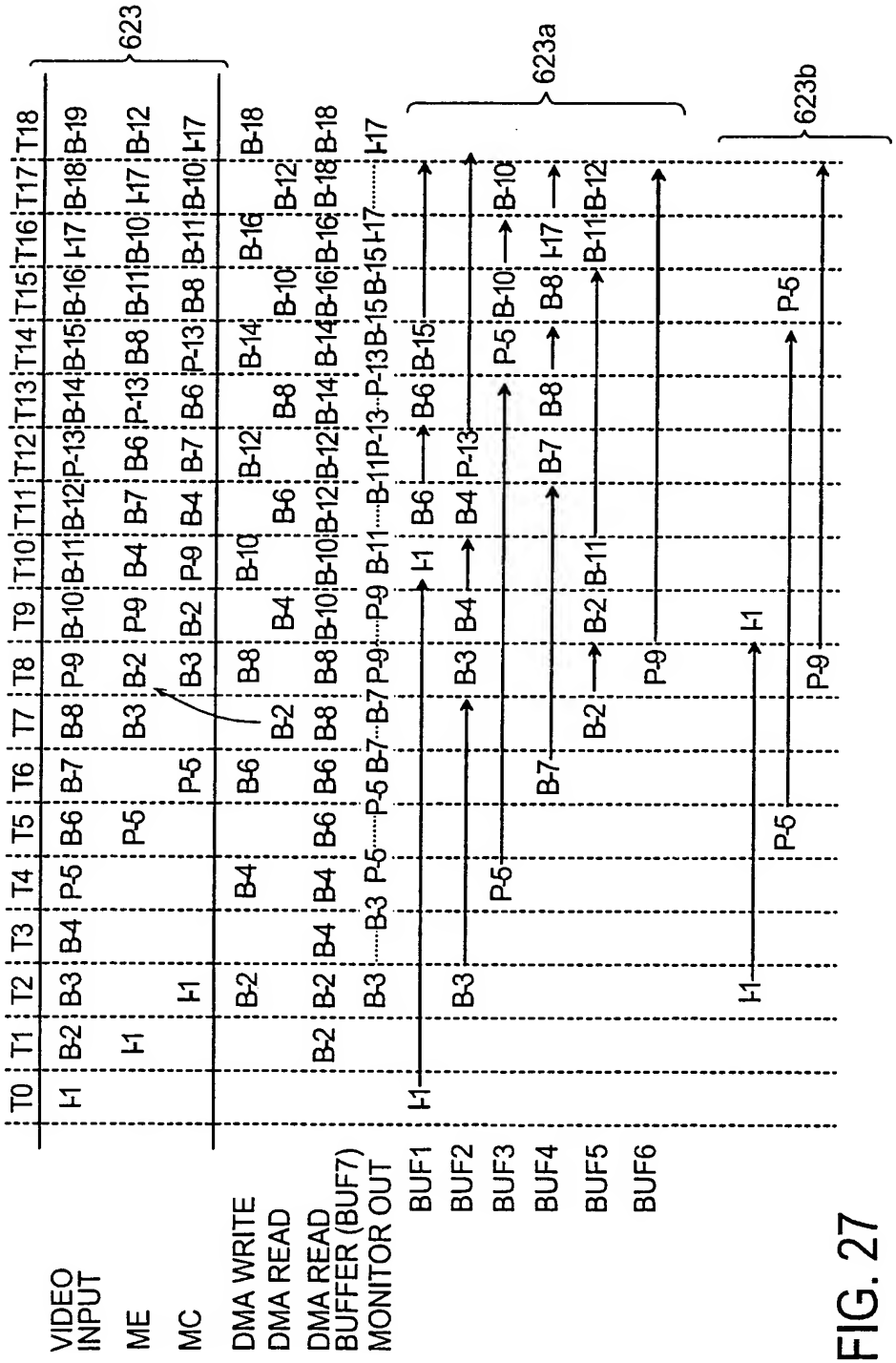

623
623a
623b
VIDEO INPUT
ME
MC
DMA WRITE
DMA READ
DMA READ BUFFER (BUF7)
MONITOR OUT
BUF1
BUF2
BUF3
BUF4
BUF5
BUF6
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18

FIG. 27

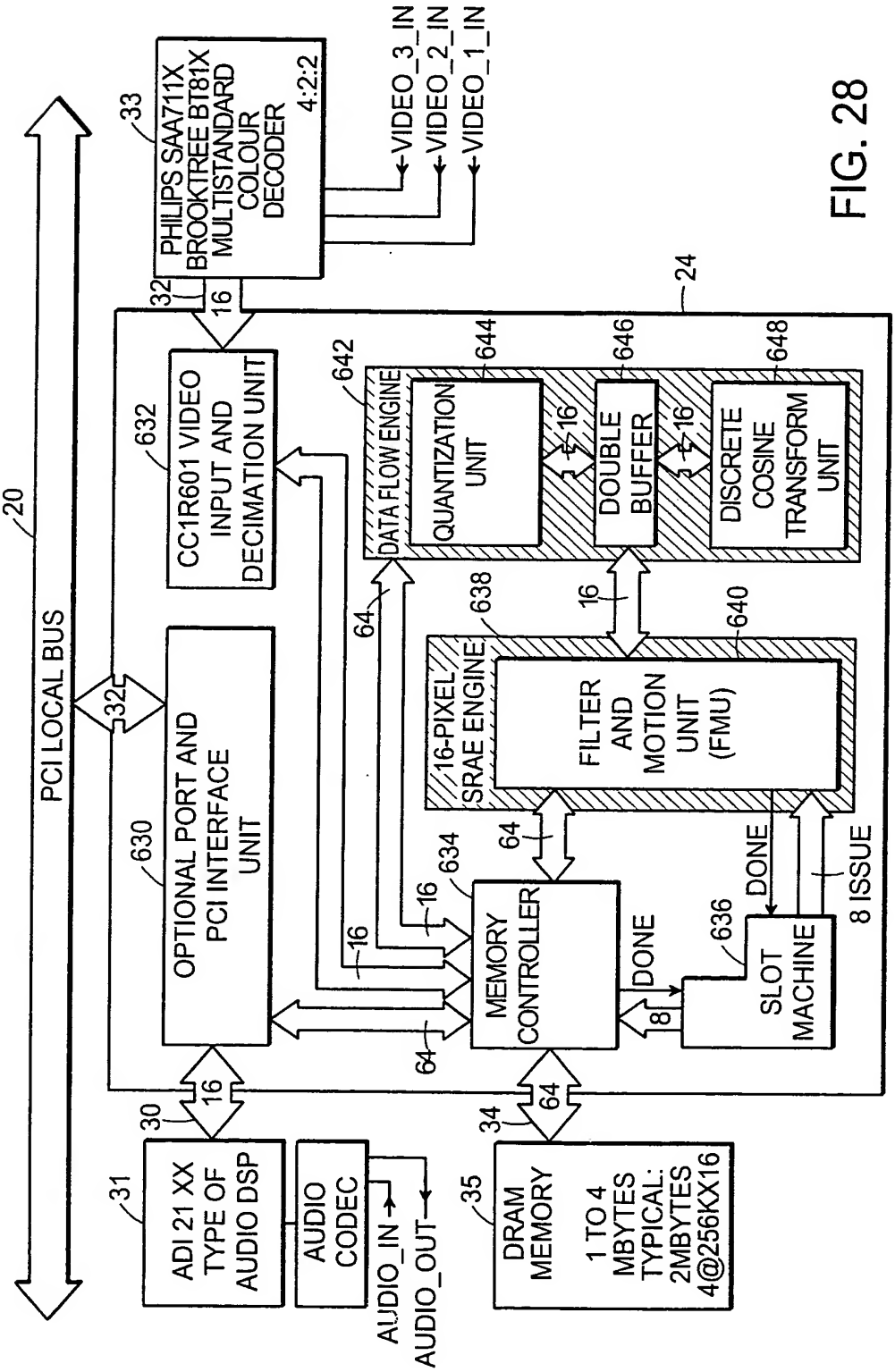

PCI LOCAL BUS
20
PHILIPS SAA711X BROOKTREE BT81X MULTISTANDARD COLOUR DECODER 4:2:2
33
VIDEO_3_IN
VIDEO_2_IN
VIDEO_1_IN
32
16
24
CC1R601 VIDEO INPUT AND DECIMATION UNIT
632
DATA FLOW ENGINE
642
QUANTIZATION UNIT
644
DOUBLE BUFFER
646
DISCRETE COSINE TRANSFORM UNIT
648
OPTIONAL PORT AND PCI INTERFACE UNIT
630
32
64
16-PIXEL SRAE ENGINE
638
FILTER AND MOTION UNIT (FMU)
640
MEMORY CONTROLLER
634
DONE
636
SLOT MACHINE
8 ISSUE
30
16
34
64
ADI 21 XX TYPE OF AUDIO DSP
31
AUDIO CODEC
AUDIO_IN
AUDIO_OUT
DRAM MEMORY
1 TO 4 MBYTES TYPICAL: 2MBYTES 4@256KX16
35

FIG. 28

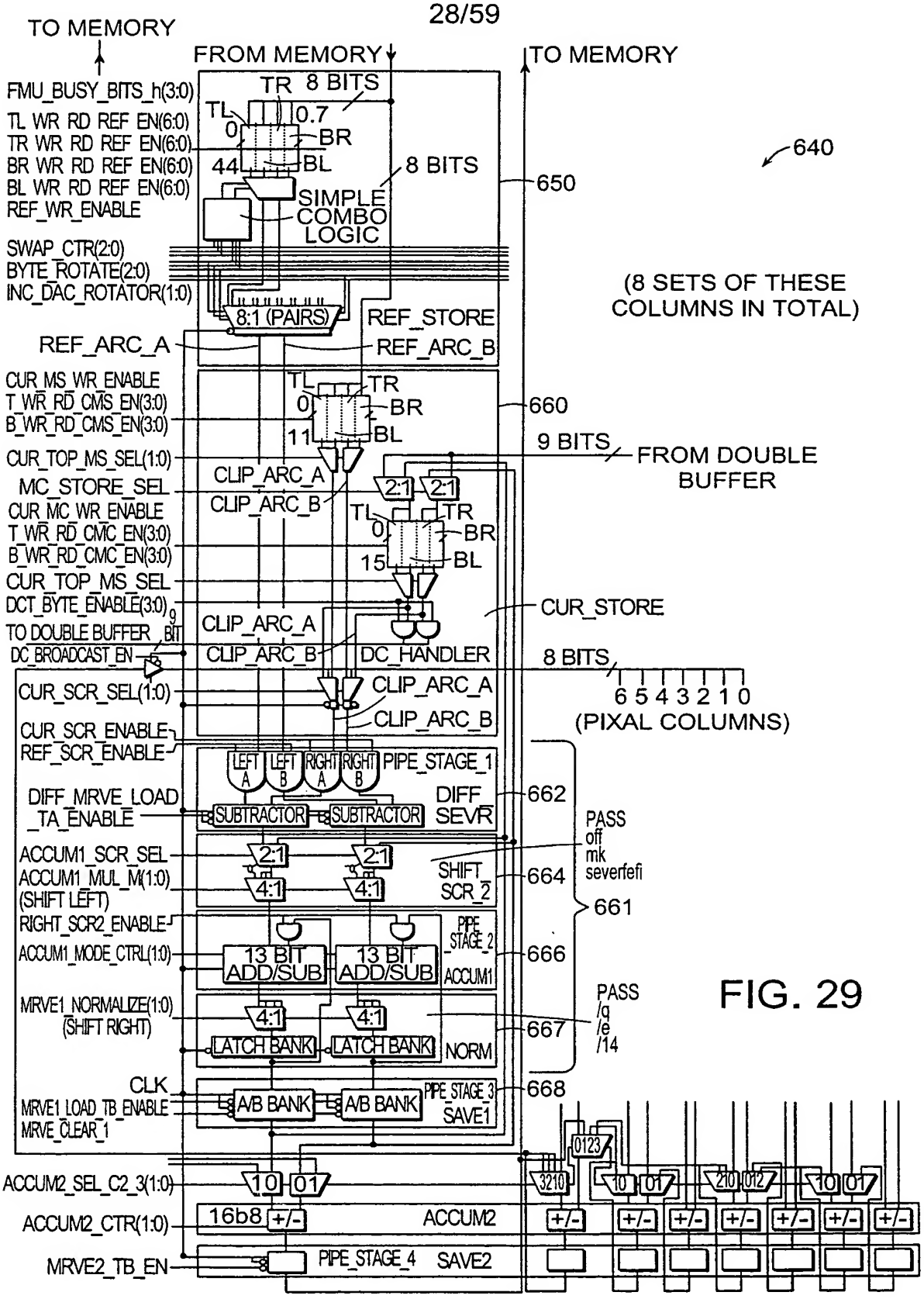

TO MEMORY
FROM MEMORY
TO MEMORY
FMU_BUSY_BITS_h(3:0)
TL WR RD REF EN(6:0)
TR WR RD REF EN(6:0)
BR WR RD REF EN(6:0)
BL WR RD REF EN(6:0)
REF_WR_ENABLE
TR 8 BITS
TL
0
0.7
BR
44
BL
8 BITS
SIMPLE COMBO LOGIC
640
650
SWAP_CTR(2:0)
BYTE_ROTATE(2:0)
INC_DAC_ROTATOR(1:0)
8:1 (PAIRS)
REF_STORE
REF_ARC_A
REF_ARC_B
(8 SETS OF THESE COLUMNS IN TOTAL)
CUR_MS_WR_ENABLE
T_WR_RD_CMS_EN(3:0)
B_WR_RD_CMS_EN(3:0)
TL
TR
0
BR
11
BL
660
9 BITS
FROM DOUBLE BUFFER
CUR_TOP_MS_SEL(1:0)
MC_STORE_SEL
CLIP_ARC_A
CLIP_ARC_B
2:1
2:1
CUR_MC_WR_ENABLE
T_WR_RD_CMC_EN(3:0)
B_WR_RD_CMC_EN(3:0)
TL
TR
0
BR
15
BL
CUR_TOP_MS_SEL
DCT_BYTE_ENABLE(3:0)
CUR_STORE
TO DOUBLE BUFFER
9 BIT
DC_BROADCAST_EN
CLIP_ARC_A
CLIP_ARC_B
DC_HANDLER
8 BITS
6 5 4 3 2 1 0
(PIXAL COLUMNS)
CUR_SCR_SEL(1:0)
CLIP_ARC_A
CLIP_ARC_B
CUR_SCR_ENABLE
REF_SCR_ENABLE
LEFT A
LEFT B
RIGHT A
RIGHT B
PIPE_STAGE_1
DIFF_MRVE_LOAD_TA_ENABLE
SUBTRACTOR
SUBTRACTOR
DIFF SEVR
662
PASS
off
mk
severfefi
ACCUM1_SCR_SEL
ACCUM1_MUL_M(1:0)
(SHIFT LEFT)
2:1
2:1
4:1
4:1
SHIFT_SCR_2
664
661
RIGHT_SCR2_ENABLE
ACCUM1_MODE_CTRL(1:0)
13 BIT ADD/SUB
13 BIT ADD/SUB
PIPE_STAGE_2
ACCUM1
666
MRVE1_NORMALIZE(1:0)
(SHIFT RIGHT)
4:1
4:1
LATCH BANK
LATCH BANK
NORM
667
PASS
/q
/e
/14
CLK
MRVE1_LOAD_TB_ENABLE
MRVE_CLEAR_1
A/B BANK
A/B BANK
PIPE_STAGE_3
SAVE1
668
0123
3210
10
01
210
012
10
01
ACCUM2_SEL_C2_3(1:0)
10
01
ACCUM2_CTR(1:0)
16b8
+/-
ACCUM2
+/-
+/-
+/-
+/-
+/-
+/-
+/-
MRVE2_TB_EN
PIPE_STAGE_4
SAVE2

FIG. 29

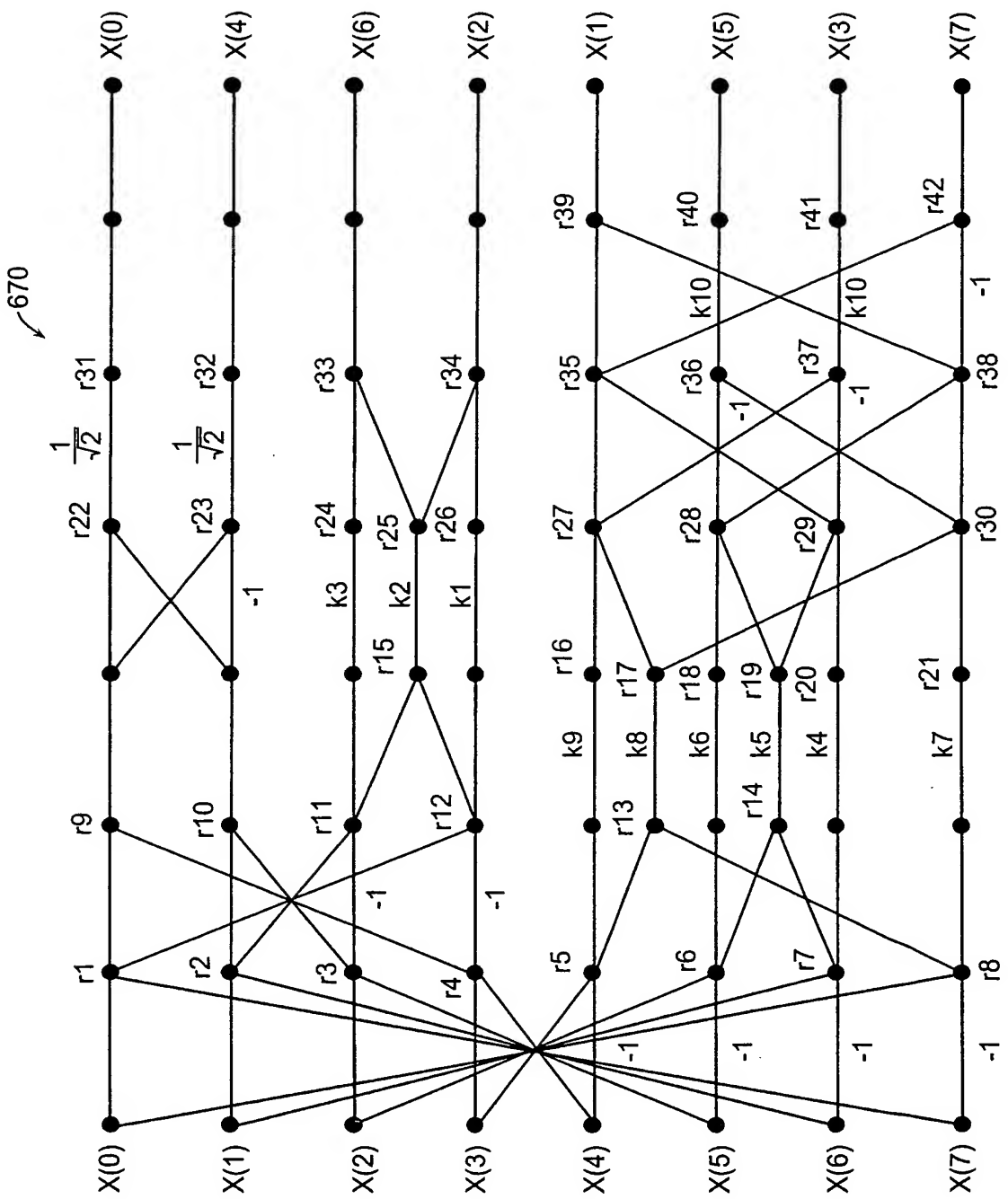

670
X(0)
X(4)
X(6)
X(2)
X(1)
X(5)
X(3)
X(7)
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13
r14
r15
r16
r17
r18
r19
r20
r21
r22
r23
r24
r25
r26
r27
r28
r29
r30
r31
r32
r33
r34
r35
r36
r37
r38
r39
r40
r41
r42
k1
k2
k3
k4
k5
k6
k7
k8
k9
k10
-1
1/√2

FIG. 30A

```
/* STAGE 1 */
        r3 = o2 + o5;
        r6 = o2 - o5;

/* STAGE 2 */
        r1 = o0 + o7;
        r8 = o0 - o7;
        r18 = r6 * 1c6;

/* STAGE 3 */
        r2 = o1 + o6;
        r7 = o1 - o6;
        r21 = r6 * 1c7;

/* STAGE 4 */
        r4 = o3 + o4;
        r14 = r5 + r7;
        r20 = r7 * 1c4;

/* STAGE 5 */
        r5 = o3 - o4;
        r11 = r2 - r3;
        r19 = r14 * 1c5;

/* STAGE 6 */
        r10 = r2 + r3;
        r16 = r5 * 1c9;

/* STAGE 7 */
        r12 = r1 - r4;
        r13 = r6 + r5;
        r24 = r11 * 1c3;

/* STAGE 8 */
        r9 = r1 + r4;
        r15 = r11 + r12;
        r26 = r12 * 1o1;

/* STAGE 9 */
        r22 = r9 + r10;
        r25 = r15 * 1c2;

/* STAGE 10 */
        r23 = r9 - r10;
        r17 = r13 * 1c8;

/* STAGE 11 */
        r28 = r18 + r19;
        r34 = r28 + r25;

/* STAGE 12 */
        r30 = r21 + r17;
        r33 = r25 + r24;
        r31 = r22 * 1c11;

/* STAGE 13 */
        r29 = r19 + r20;
        r36 = r30 - r28;
        r32 = r23 * 1c11;

/* STAGE 14 */
        r27 = r16 + r17;
        r36 = r28 + r30;
        r40 = r35 * 1c10;

/* STAGE 15 */
        r35 = r27 + r29;

/* STAGE 16 */
        r37 = r27 - r29;
        r39 = r35 + r38;

/* STAGE 17 */
        r42 = r35 - r38;
        r41 = r37 * 1c10;

/* STAGE 18 */
        r41 = r37 * 1c10;
```

672

FIG. 30B

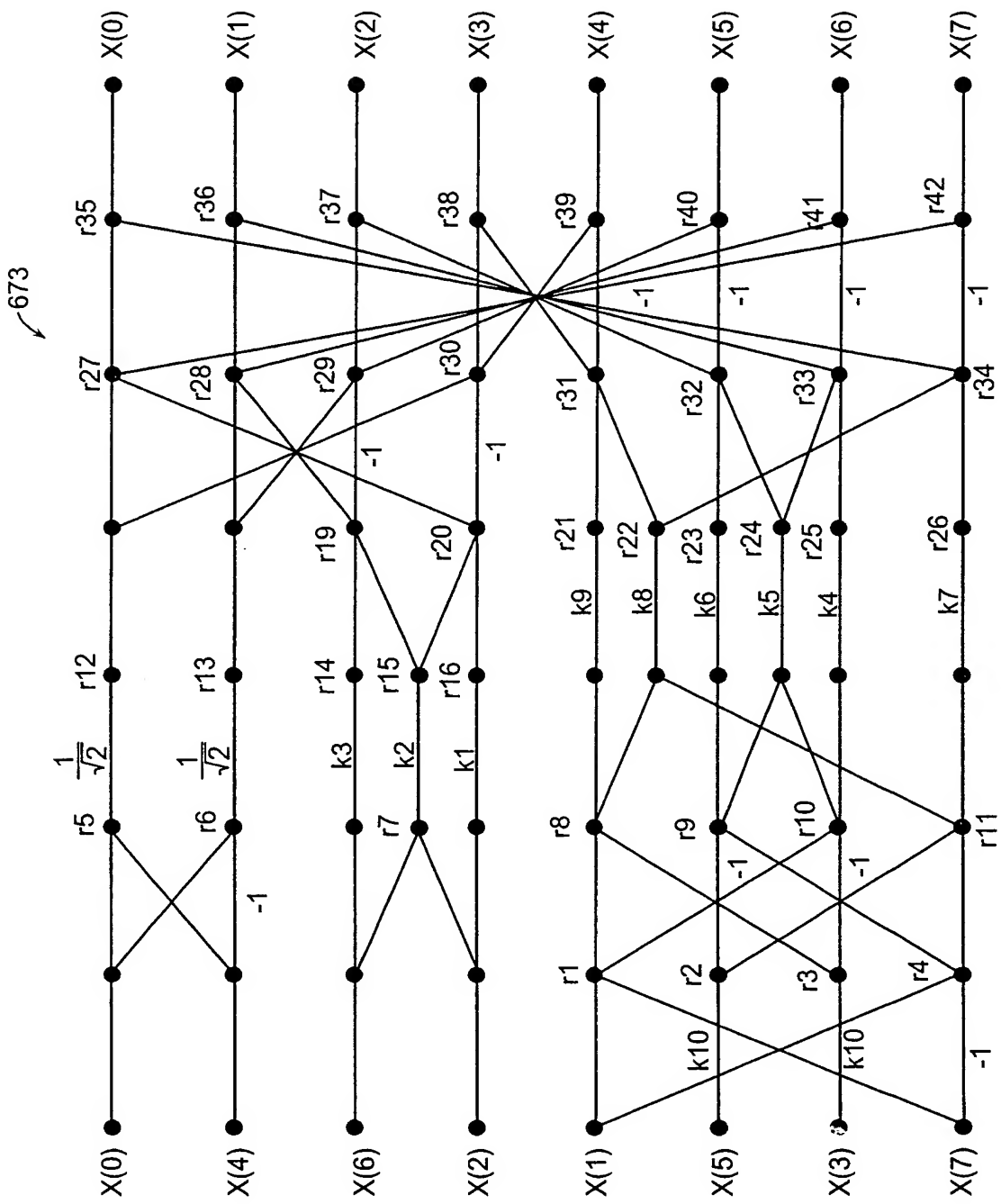

673
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)
X(0)
X(4)
X(6)
X(2)
X(1)
X(5)
X(3)
X(7)
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13
r14
r15
r16
r19
r20
r21
r22
r23
r24
r25
r26
r27
r28
r29
r30
r31
r32
r33
r34
r35
r36
r37
r38
r39
r40
r41
r42
k1
k2
k3
k4
k5
k6
k7
k8
k9
k10
1/√2
1/√2
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1

FIG. 30C

674

```
/* STAGE 1 */
        r2 = o5 * 1c10;
/* STAGE 2 */
        r5 = o0 + o4;
        r6 = o0 - o4;
        r3 = o3 * 1c10;
/* STAGE 3 */
        r1 = o1 + o7;
        r4 = o1 - o7;
        r13 = r6 * 1c11;
/* STAGE 4 */
        r7 = o6 + o2;
        r8 = r1 + r3;
        r14 = o6 * 1c3;
/* STAGE 5 */
        r11 = r2 + r4;
        r10 = r1 - r3;
        r15 = r7 * 1c2;
/* STAGE 6 */
        r9 = r4 - r2;
        r17 = r11 + r8
        r16 = o2 * 1c1;
/* STAGE 7 */
        r18 = r9 + r10;
        r25 = r10 * 1c4;
/* STAGE 8 */
        r20 = r15 + r16;
        r24 = r18 * 1c5;
/* STAGE 9 */
        r19 = r14 + r15;
        r23 = r9 * 1c6;
/* STAGE 10 */
        r28 = r19 + r13;
        r33 = r25 +r24;
        r21 = r8 * 1c9;
/* STAGE 11 */
        r29 = r13 - r19;
        r36 = r28 + r33;
        r22 = r17 * 1c8;
/* STAGE 12 */
        r32 = r23 + r24;
        r41 = r28 - r33;
        r12 = r5 * 1c11;
/* STAGE 13 */
        r31 = r21 + r22;
        r37 = r29 + r32;
        r26 = r11 * 1c7;
/* STAGE 14 */
        r30 = r12 - r20;
        r40 = r29 - r32;
/* STAGE 15 */
        r27 = r20 + r12;
        r38 + r30 + r31;
/* STAGE 16 */
        r34 = r22 + r26;
        r39 = R30 - r31;
/* STAGE 17 */
        r35 = r27 + r34;
/* STAGE 18 */
        r42 = r27 - r34;
```

FIG. 30D

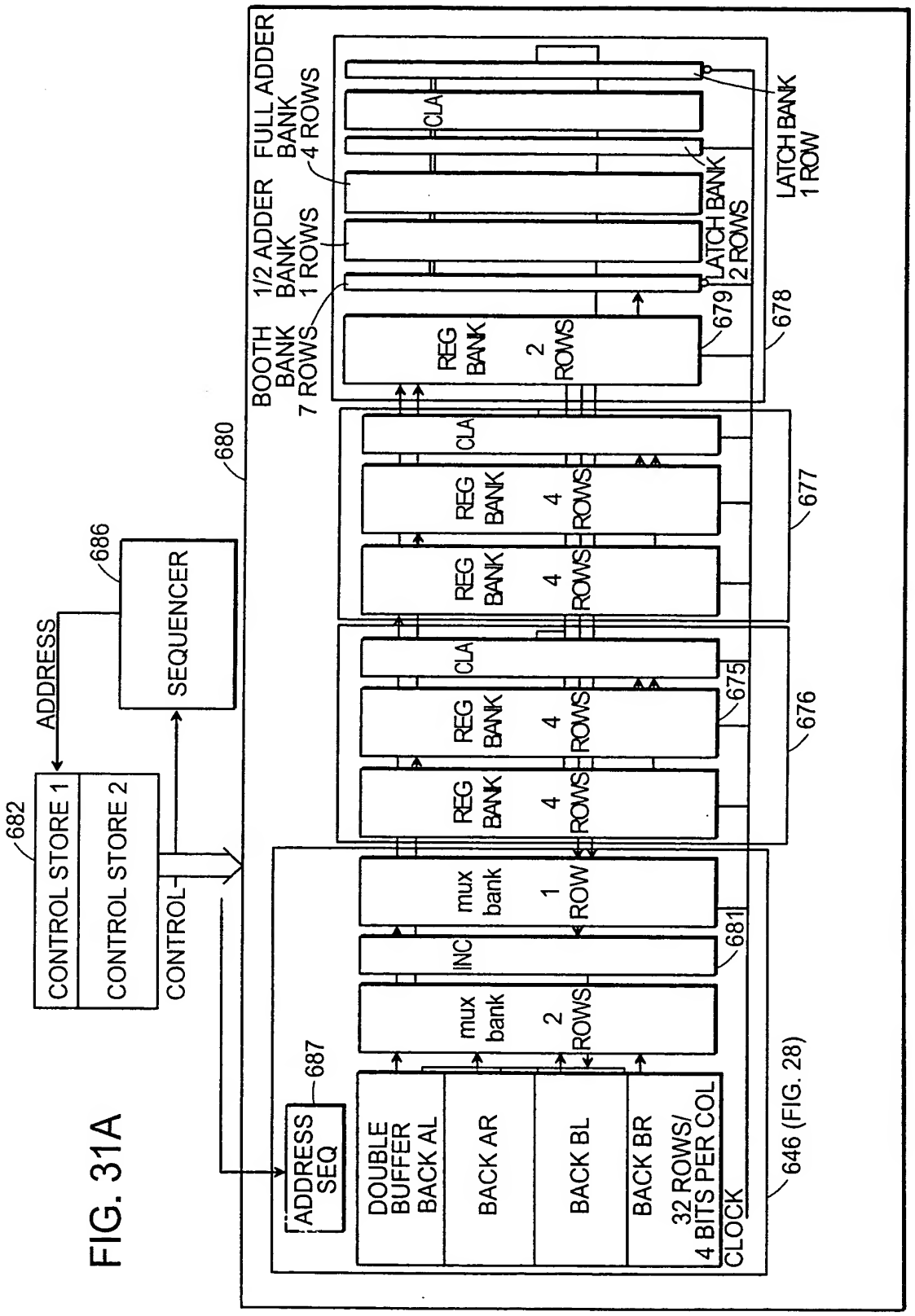

680
686
SEQUENCER
ADDRESS
682
CONTROL STORE 1
CONTROL STORE 2
CONTROL
BOOTH BANK 7 ROWS
1/2 ADDER BANK 1 ROWS
FULL ADDER BANK 4 ROWS
CLA
REG BANK 2 ROWS
679
678
LATCH BANK 2 ROWS
LATCH BANK 1 ROW
677
REG BANK 4 ROWS
676
675
mux bank 2 ROWS
INC
mux bank 1 ROW
681
687
ADDRESS SEQ
DOUBLE BUFFER BACK AL
BACK AR
BACK BL
BACK BR
32 ROWS/ 4 BITS PER COL
CLOCK
646 (FIG. 28)

FIG. 31A

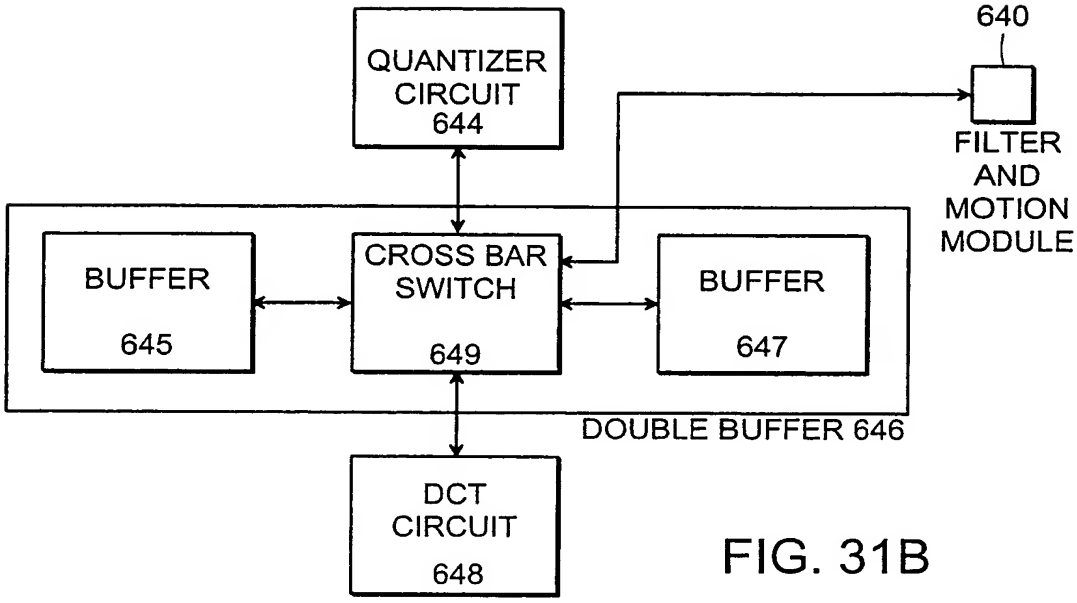


FIG. 31B

BLOCK ELEMENT ARRAY:

	LEFT OPERAND ROW TRANSFORM				RIGHT OPERAND ROW TRANSFORM				
	0	1	2	3	4	5	6	7	ROW TRANSFORM
LEFT OPERAND COLUMN TRANSFORM	8	9	10	11	12	13	14	15	
	16	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	31	
RIGHT OPERAND COLUMN TRANSFORM	32	33	34	35	36	37	38	39	
	40	41	42	43	44	45	46	47	
	48	49	50	51	52	53	54	55	
	56	57	58	59	60	61	62	63	
COLUMN TRANSFORM									

FIG. 32

DCT DOUBLE BUFFER STORAGE:

DCT OPERAND FETCH ORDER: (654)

LEFT	RIGHT
2	5
0	7
1	6
3	4

→

DCT DATA STORE ORGANIZATION (651)

RAM ADDR WORDLINE				
0	63	0	7	56
1	62	1	6	57
2	61	2	5	58
3	60	3	4	59
4	55	8	15	48
5	54	9	14	49
6	53	10	13	50
7	52	11	12	51
8	47	16	23	40
9	46	17	22	41
10	45	18	21	42
11	44	19	20	43
12	39	24	31	32
13	38	25	30	33
14	37	26	29	34
15	36	27	28	35

DCT CONDITION / RAM ADDR BITLINE CONTROL (653)

ROW 0:3		L	R	
ROW 4:7	R			L
COL 0:3		L		R
COL 4:7	R		L	

→

DCT DOUBLE BUFFER ADDRESSING (652)

RAM ADDR WORDLINE	TRANSFORM DOMAIN	ALGORITHM
2, 0, 1, 3	ROWS 0:3	ROW 0:3
14, 12, 13, 15	ROWS 4:7	ROW 0:3 +12
8, 0, 4, 12	COLS 0:3	ROW 0:3 x 4
11, 3, 7, 15	COLS 4:7	ROW 0:3 x 4 +3

FIG. 33

IDCT DOUBLE BUFFER STORAGE:

IDCT OPERAND FETCH ORDER:

LEFT	RIGHT
3	5
0	4
1	7
2	6

→

RAM ADDR WORDLINE	IDCT DATA STORE ORGANIZATION			
0	36	0	4	32
1	39	1	7	33
2	38	2	6	34
3	37	3	5	35
4	60	8	12	56
5	63	9	15	57
6	62	10	14	58
7	61	11	13	59
8	52	16	20	48
9	55	17	23	49
10	54	18	22	50
11	53	19	21	51
12	44	24	28	40
13	47	25	31	41
14	46	26	30	42
15	45	27	29	43

IDCT CONDITION	RAM ADDR BITLINE CONTROL			
ROW 0:3		L	R	
ROW 4:7	R			L
COL 0:3		L		R
COL 4:7	R		L	

IDCT DOUBLE BUFFER ADDRESSING

RAM ADDR WORDLINE	TRANSFORM DOMAIN	ALGORITHM
3 0 1 2	ROWS 0:3	ROW 0:3
3 0 1 2	ROWS 4:7	ROW 0:3
12 0 4 8	COLS 0:3	ROW 0:3 x 4
12 0 4 8	COLS 4:7	ROW 0:3 x 4

FIG. 34

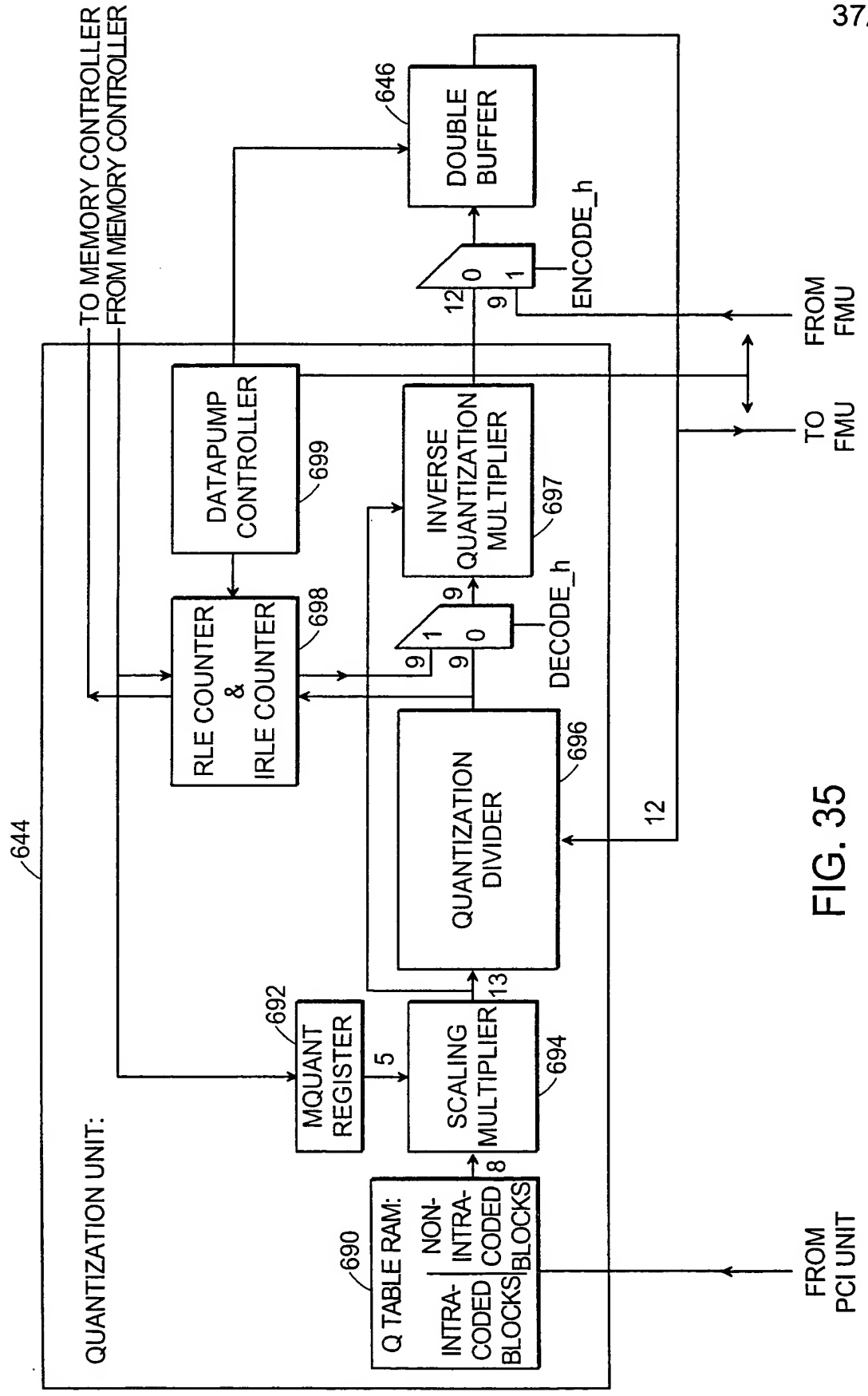

644
QUANTIZATION UNIT:
TO MEMORY CONTROLLER
FROM MEMORY CONTROLLER
690
Q TABLE RAM:
INTRA-CODED BLOCKS
NON-INTRA-CODED BLOCKS
8
692
MQUANT REGISTER
5
SCALING MULTIPLIER
694
13
QUANTIZATION DIVIDER
696
12
RLE COUNTER & IRLE COUNTER
698
9
9
1
0
DECODE_h
INVERSE QUANTIZATION MULTIPLIER
697
DATAPUMP CONTROLLER
699
646
DOUBLE BUFFER
12
9
1
0
ENCODE_h
FROM FMU
TO FMU
FROM PCI UNIT

FIG. 35

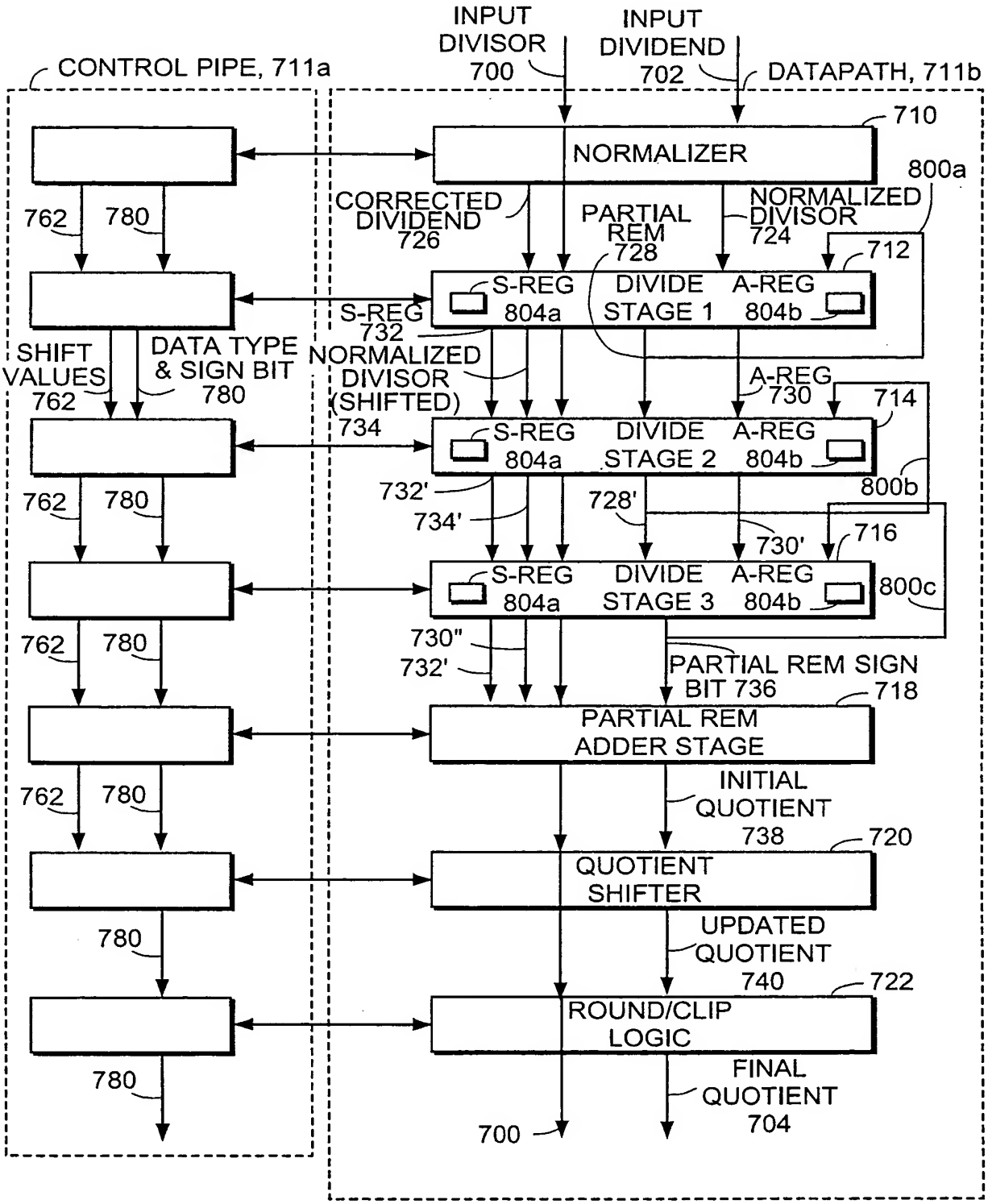

INPUT DIVISOR 700
INPUT DIVIDEND 702
CONTROL PIPE, 711a
DATAPATH, 711b
710
NORMALIZER
800a
762
780
CORRECTED DIVIDEND 726
PARTIAL REM 728
NORMALIZED DIVISOR 724
712
S-REG 804a
DIVIDE STAGE 1
A-REG 804b
S-REG 732
SHIFT VALUES 762
DATA TYPE & SIGN BIT 780
NORMALIZED DIVISOR (SHIFTED) 734
A-REG 730
714
S-REG 804a
DIVIDE STAGE 2
A-REG 804b
800b
732'
728'
734'
730'
716
S-REG 804a
DIVIDE STAGE 3
A-REG 804b
800c
730"
732'
PARTIAL REM SIGN BIT 736
718
PARTIAL REM ADDER STAGE
INITIAL QUOTIENT 738
720
QUOTIENT SHIFTER
UPDATED QUOTIENT 740
722
ROUND/CLIP LOGIC
FINAL QUOTIENT 704
700

FIG. 36

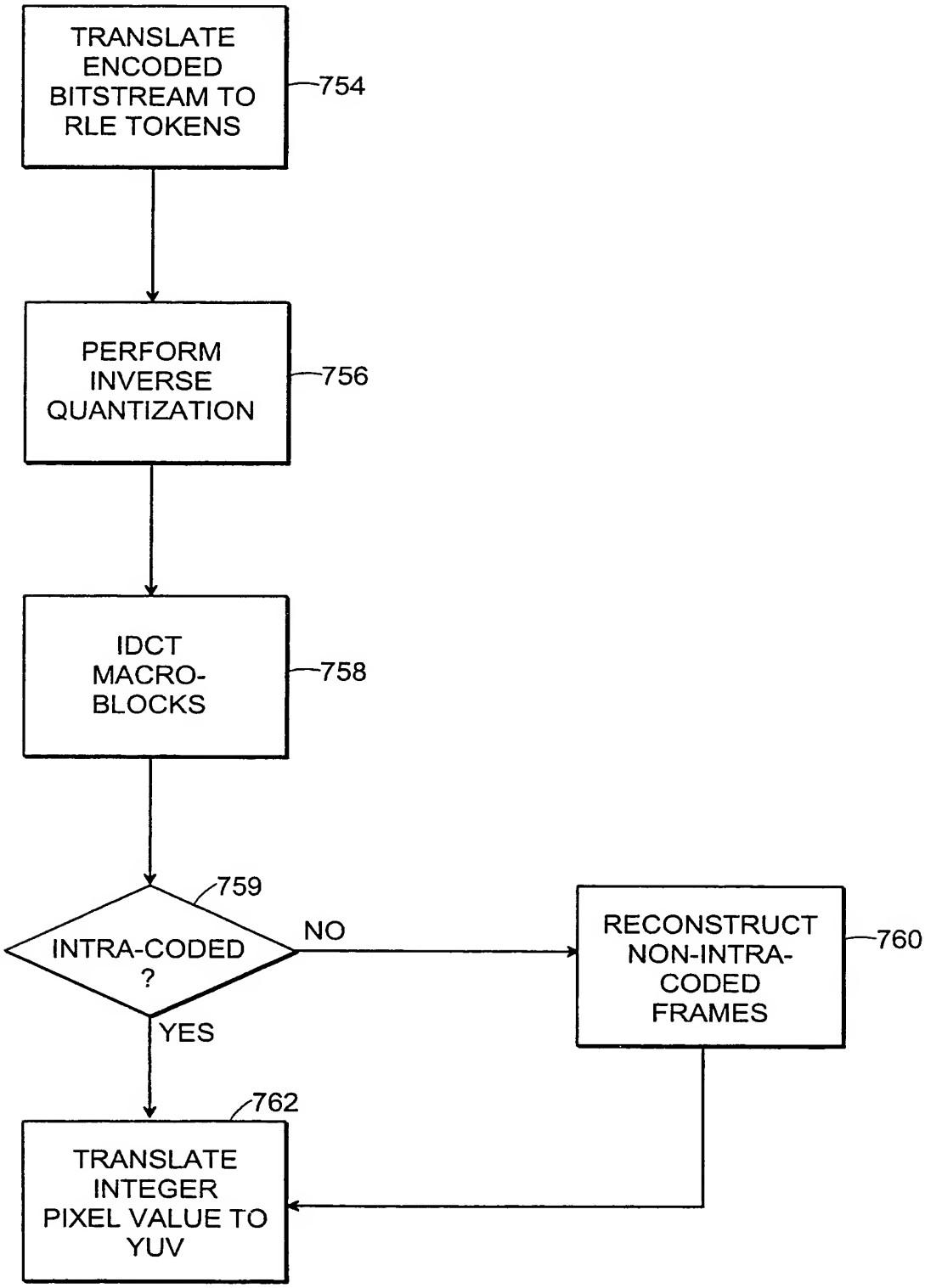

TRANSLATE ENCODED BITSTREAM TO RLE TOKENS
754
PERFORM INVERSE QUANTIZATION
756
IDCT MACRO-BLOCKS
758
759
INTRA-CODED ?
NO
YES
RECONSTRUCT NON-INTRA-CODED FRAMES
760
762
TRANSLATE INTEGER PIXEL VALUE TO YUV

FIG. 37

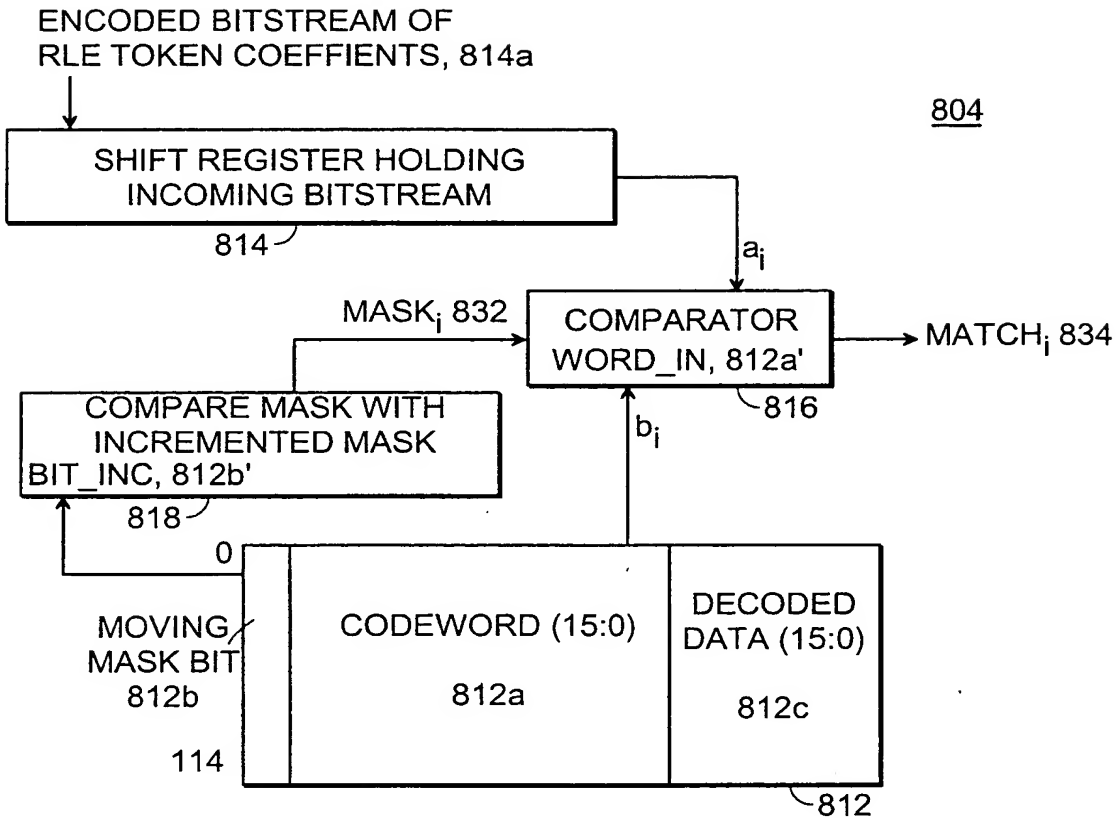

ENCODED BITSTREAM OF
RLE TOKEN COEFFIENTS, 814a
804
SHIFT REGISTER HOLDING
INCOMING BITSTREAM
814
a_i
$MASK_i$ 832
COMPARATOR
WORD_IN, 812a'
$MATCH_i$ 834
816
COMPARE MASK WITH
INCREMENTED MASK
BIT_INC, 812b'
818
b_i
0
MOVING
MASK BIT
812b
114
CODEWORD (15:0)
812a
DECODED
DATA (15:0)
812c
812

FIG. 38

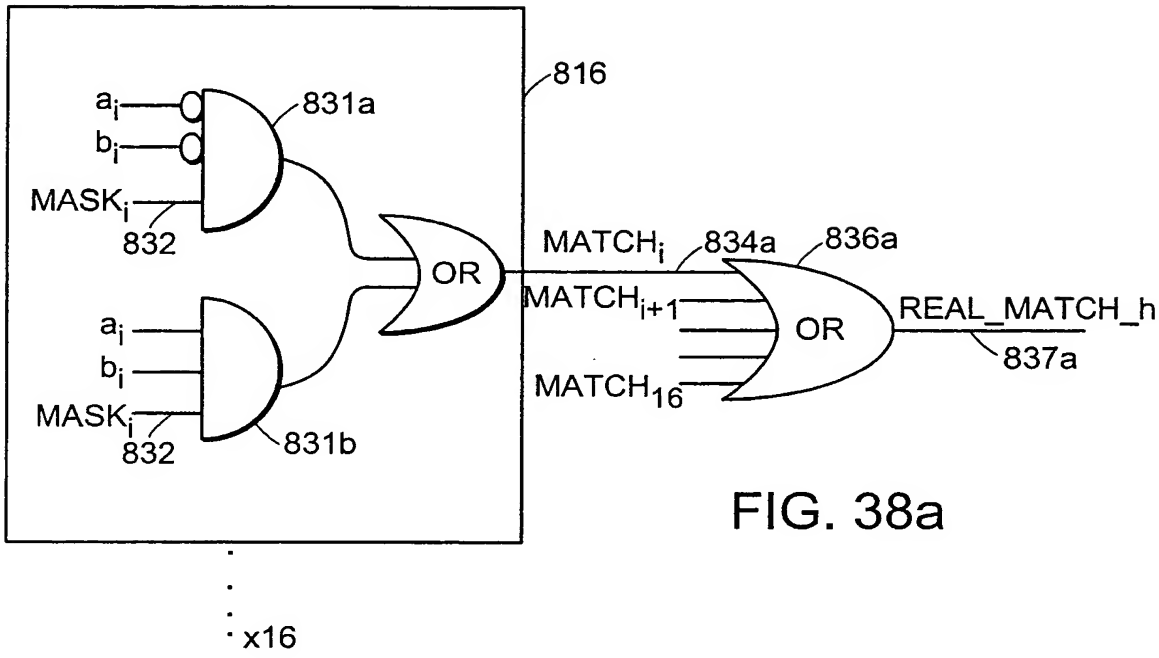


FIG. 38a

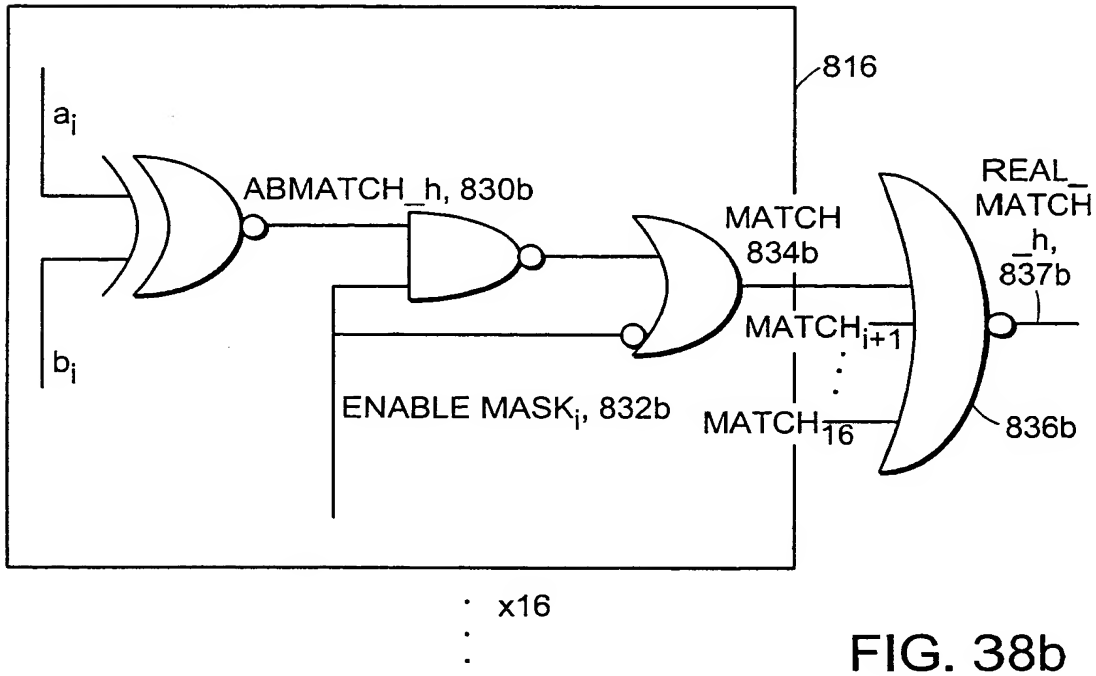


FIG. 38b

ENCODED INCOMING BITSTREAM 814a

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
1	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0

FIG. 39

CODE WORD TABLE 812

MOVING MASK BIT 812b	VARIABLE LENGTH CODE (812a) NOTE1	DECODED DATA (RUN) 812c	DECODED DATA (LEVEL) 812c
0	10	END OF BLOCK	
0	1 s(NOTE2)	0	1
1	11 s (NOTE3)	0	1
1	011 s	1	1
1	0100 s	0	2
0	0101 s	2	1
1	0010 1 s	0	3
0	0011 1 s	3	1
0	0011 0 s	4	1
1	0001 10 s	1	2
0	0001 11 s	5	1
0	0001 01 s	6	1
0	0001 00 s	7	1
0	0000 01 s	ESCAPE	
1	0000 110 s	0	4
0	0000 100 s	2	2
0	0000 111 s	8	1
0	0000 101 s	9	1
1	0010 0110 s	0	5
0	0010 0001 s	0	6
0	0010 0101 s	1	3
0	0010 0100 s	3	2
0	0010 0111 s	10	1
0	0010 0011 s	11	1
0	0010 0010 s	12	1
0	0010 0000 s	13	1
1	0010 0000 s	13	1
0	0000 0010 10 s	0	7
0	0000 0011 00 s	1	4
0	0000 0010 11 s	2	3
0	0000 0011 11 s	4	2
0	0000 0010 01 s	5	2
0	0000 0011 10 s	14	1
0	0000 0011 01 s	15	1
0	0000 0010 00 s	16	1

NOTE1-THE LAST BIT 's' DENOTES THE SIGN OF THE LEVEL, '0' FOR POSITVE, '1' FOR NEGATIVE.
NOTE2-THIS CODE SHALL BE USED FOR THE FIRST (DC) COEFFICIENT IN THE BLOCK
NOTE3-THIS CODE SHALL BE USED FOR ALL OTHER COEFFICIENTS.

FIG. 39a

CODE WORD TABLE 812 (CONTINUED)

MOVING MASK BIT 812b	VARIABLE LENGTH CODE (812a)	DECODED DATA (RUN) 812c	DECODED DATA (LEVEL) 812c
0	0000 0001 1101 s	0	8
0	0000 0001 1000 s	0	9
0	0000 0001 0011 s	0	10
0	0000 0001 0000 s	0	11
0	0000 0001 1011 s	1	5
0	0000 0001 0100 s	2	4
0	0000 0001 1100 s	3	3
0	0000 0001 0010 s	4	3
0	0000 0001 1110 s	6	2
0	0000 0001 0101 s	7	2
0	0000 0001 0001 s	8	2
0	0000 0001 1111 s	17	1
0	0000 0001 1010 s	18	1
0	0000 0001 1001 s	19	1
0	0000 0001 0111 s	20	1
0	0000 0001 0110 s	21	1
1	0000 0000 1101 0 s	0	12
0	0000 0000 1100 1 s	0	13
0	0000 0000 1100 0 s	0	14
0	0000 0000 1011 1 s	0	15
0	0000 0000 1011 0 s	1	6
0	0000 0000 1010 1 s	1	7
0	0000 0000 1010 0 s	2	5
0	0000 0000 1001 1 s	3	4
0	0000 0000 1001 0 s	5	3
0	0000 0000 1000 1 s	9	2
0	0000 0000 1000 0 s	10	2
0	0000 0000 1111 1 s	22	1
0	0000 0000 1111 0 s	23	1
0	0000 0000 1110 1 s	24	1
0	0000 0000 1110 0 s	25	1
0	0000 0000 1101 1 s	26	1
NOTE-THE LAST BIT 'S' DENOTES THE SIGN OF THE LEVEL, '0' FOR POSITIVE, '1' FOR NEGATIVE.			

FIG. 39b

CODE WORD TABLE 812 (CONTINUED)

MOVING MASK BIT 812b	VARIABLE LENGTH CODE (812a)	DECODED DATA (RUN) 812c	DECODED DATA (LEVEL) 812c
1	0000 0000 0111 11 s	0	16
0	0000 0000 0111 10 s	0	17
0	0000 0000 0111 01 s	0	18
0	0000 0000 0111 00 s	0	19
0	0000 0000 0110 11 s	0	20
0	0000 0000 0110 10 s	0	21
0	0000 0000 0110 01 s	0	22
0	0000 0000 0110 00 s	0	23
0	0000 0000 0101 11 s	0	24
0	0000 0000 0101 10 s	0	25
0	0000 0000 0101 01 s	0	26
0	0000 0000 0101 00 s	0	27
0	0000 0000 0100 11 s	0	28
0	0000 0000 0100 10 s	0	29
0	0000 0000 0100 01 s	0	30
0	0000 0000 0100 00 s	0	31
1	0000 0000 0011 000 s	0	32
0	0000 0000 0010 111 s	0	33
0	0000 0000 0010 110 s	0	34
0	0000 0000 0010 101 s	0	35
0	0000 0000 0010 100 s	0	36
0	0000 0000 0010 011 s	0	37
0	0000 0000 0010 010 s	0	38
0	0000 0000 0010 001 s	0	39
0	0000 0000 0010 000 s	0	40
0	0000 0000 0011 111 s	1	8
0	0000 0000 0011 110 s	1	9
0	0000 0000 0011 101 s	1	10
0	0000 0000 0011 100 s	1	11
0	0000 0000 0011 011 s	1	12
0	0000 0000 0011 010 s	1	13
0	0000 0000 0011 001 s	1	14
NOTE-THE LAST BIT 'S' DENOTES THE SIGN OF THE LEVEL, '0' FOR POSITIVE, '1' FOR NEGATIVE.			

FIG. 39c

CODE WORD TABLE 812 (CONCLUDED)

MOVING MASK BIT 812b	VARIABLE LENGTH CODE (812a)	DECODED DATA (RUN) 812c	DECODED DATA (LEVEL) 812c
1	0000 0000 0001 0011 s	1	15
0	0000 0000 0001 0010 s	1	16
0	0000 0000 0001 0001 s	1	17
0	0000 0000 0001 0000 s	1	18
0	0000 0000 0001 0100 s	6	3
0	0000 0000 0001 1010 s	11	2
0	0000 0000 0001 1001 s	12	2
0	0000 0000 0001 1000 s	13	2
0	0000 0000 0001 0111 s	14	2
0	0000 0000 0001 0110 s	15	2
0	0000 0000 0001 0101 s	16	2
0	0000 0000 0001 1111 s	27	1
0	0000 0000 0001 1110 s	28	1
0	0000 0000 0001 1101 s	29	1
0	0000 0000 0001 1100 s	30	1
0	0000 0000 0001 1011 s	31	1
NOTE-THE LAST BIT 'S' DENOTES THE SIGN OF THE LEVEL, '0' FOR POSITIVE, '1' FOR NEGATIVE.			

FIG. 39d

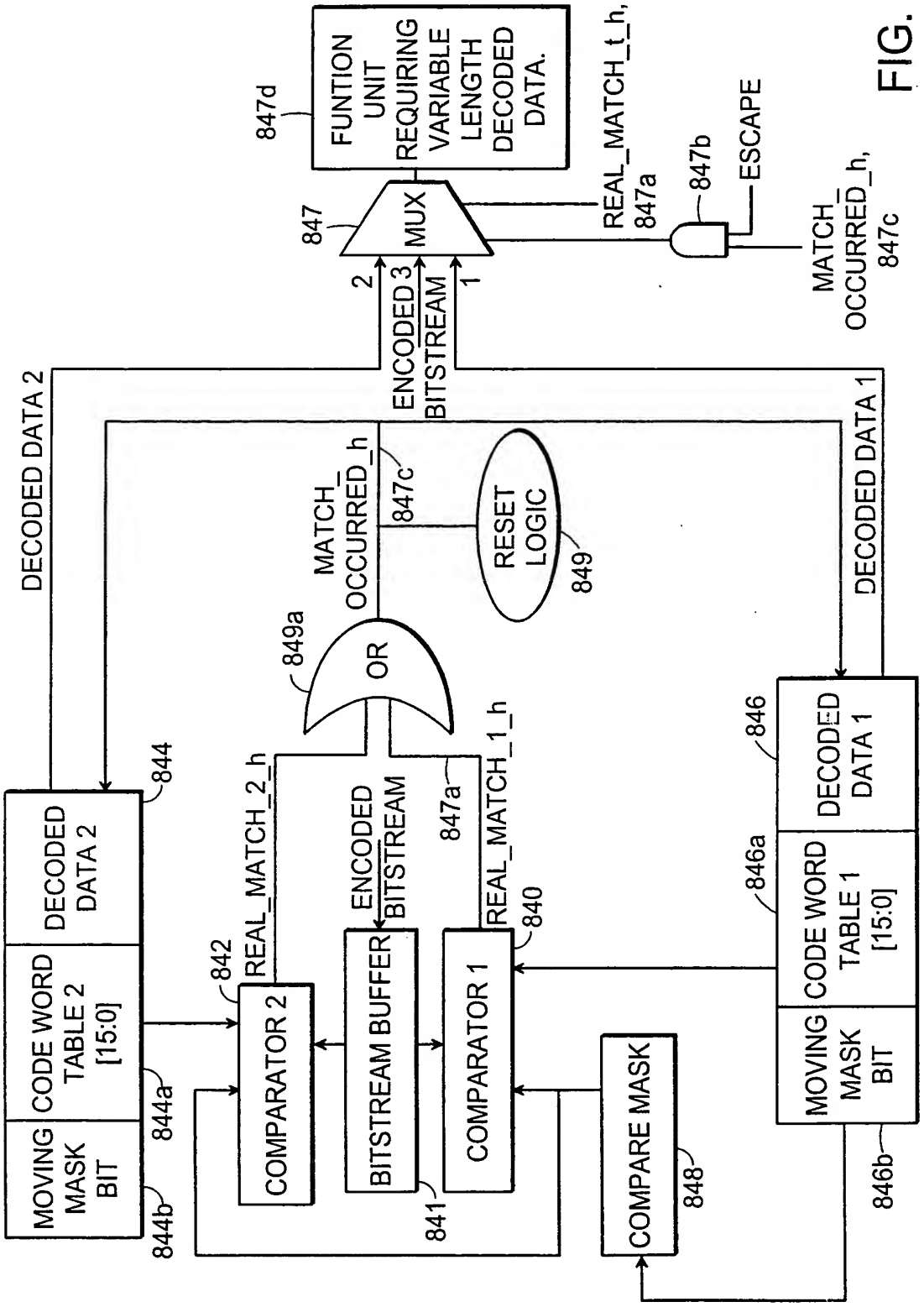

847d
FUNTION UNIT REQUIRING VARIABLE LENGTH DECODED DATA.
847
MUX
REAL_MATCH_t_h, 847a
847b
ESCAPE
MATCH_ OCCURRED_h, 847c
2
ENCODED BITSTREAM 3
1
DECODED DATA 2
DECODED DATA 1
MATCH_ OCCURRED_h
847c
RESET LOGIC
849
849a
OR
844
DECODED DATA 2
CODE WORD TABLE 2 [15:0]
MOVING MASK BIT
844a
844b
842
REAL_MATCH_2_h
COMPARATOR 2
ENCODED BITSTREAM
BITSTREAM BUFFER
841
847a
REAL_MATCH_1_h
840
COMPARATOR 1
COMPARE MASK
848
846
DECODED DATA 1
846a
CODE WORD TABLE 1 [15:0]
MOVING MASK BIT
846b

FIG. 40

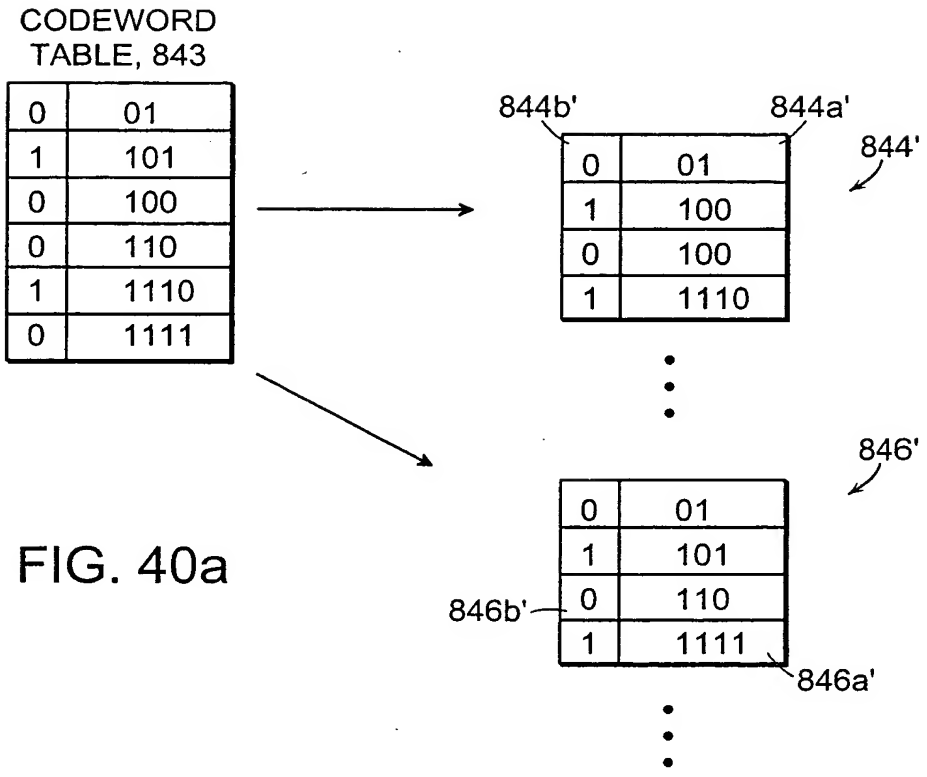

CODEWORD TABLE, 843
0 01
1 101
0 100
0 110
1 1110
0 1111
844b'
844a'
844'
0 01
1 100
0 100
1 1110
846'
0 01
1 101
0 110
1 1111
846b'
846a'

FIG. 40a

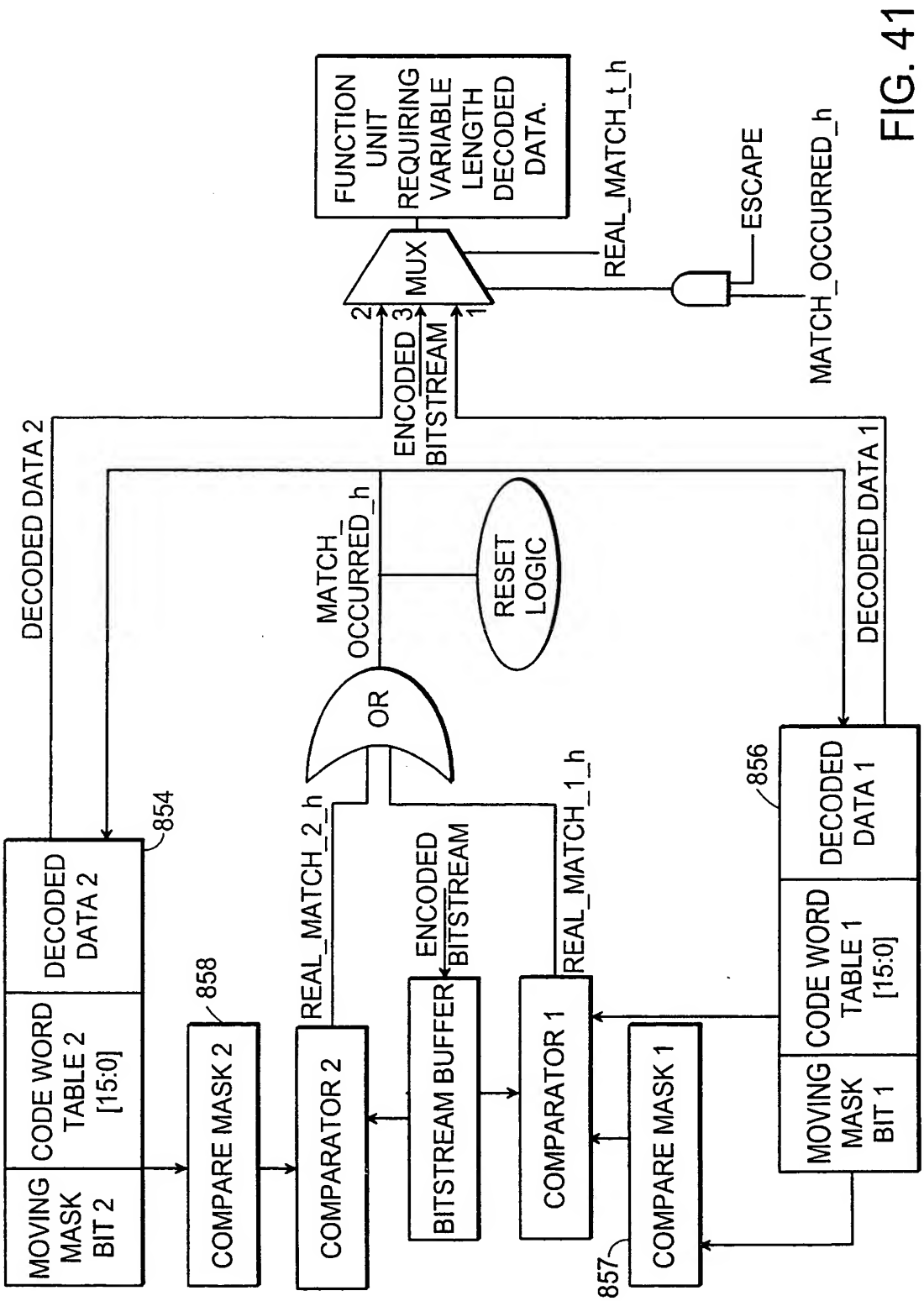

FUNCTION UNIT REQUIRING VARIABLE LENGTH DECODED DATA.
MUX
2
3
1
REAL_MATCH_t_h
ESCAPE
MATCH_OCCURRED_h
ENCODED BITSTREAM
DECODED DATA 2
DECODED DATA 1
MATCH_ OCCURRED_h
RESET LOGIC
OR
REAL_MATCH_2_h
REAL_MATCH_1_h
ENCODED BITSTREAM
854
856
858
857
MOVING MASK BIT 2
CODE WORD TABLE 2 [15:0]
DECODED DATA 2
COMPARE MASK 2
COMPARATOR 2
BITSTREAM BUFFER
COMPARATOR 1
COMPARE MASK 1
MOVING MASK BIT 1
CODE WORD TABLE 1 [15:0]
DECODED DATA 1

FIG. 41

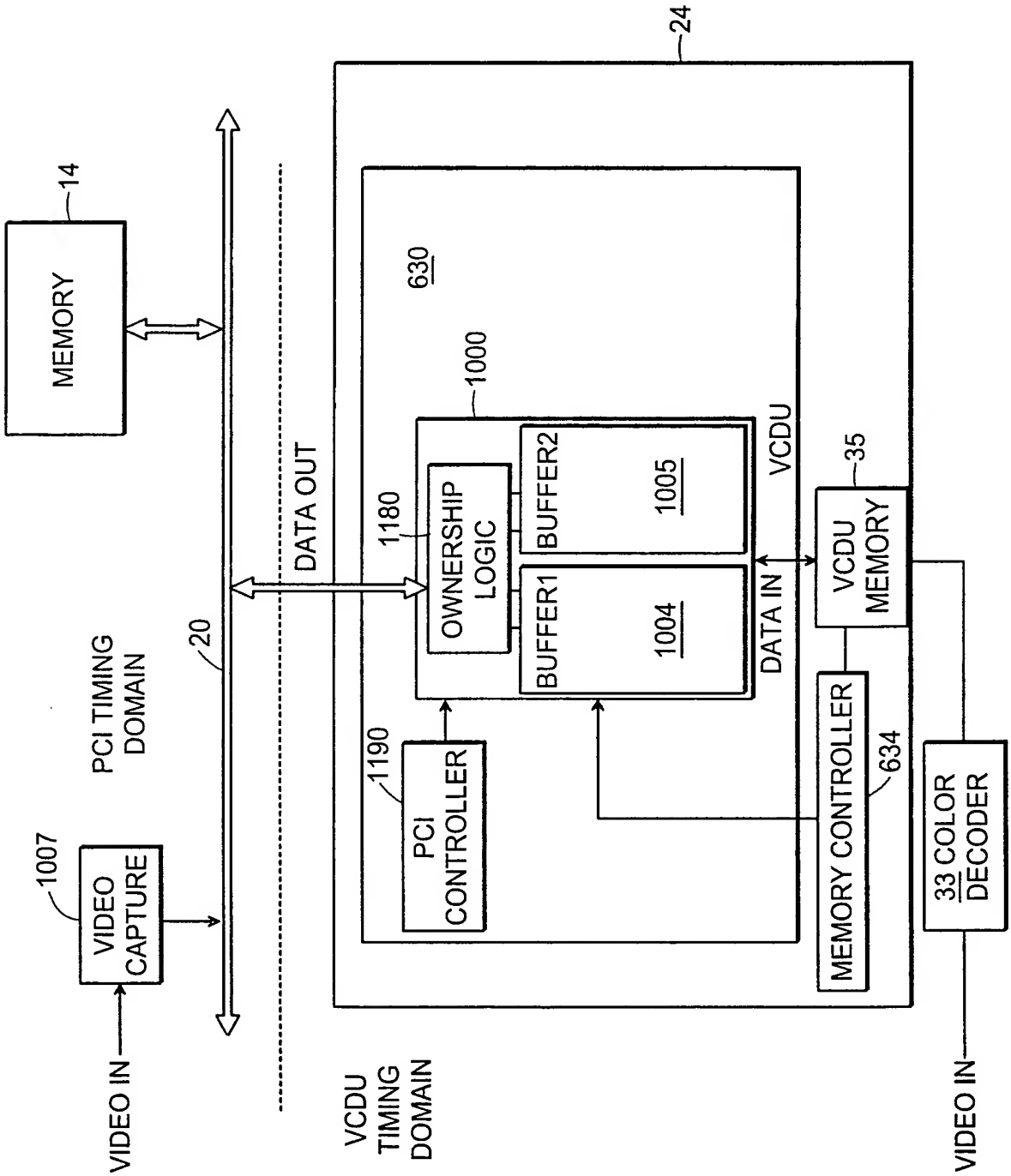

MEMORY
14
20
PCI TIMING DOMAIN
DATA OUT
24
630
1000
1180
OWNERSHIP LOGIC
BUFFER2
1005
BUFFER1
1004
VCDU
DATA IN
35
VCDU MEMORY
1190
PCI CONTROLLER
MEMORY CONTROLLER
634
33 COLOR DECODER
1007
VIDEO CAPTURE
VIDEO IN
VCDU TIMING DOMAIN
VIDEO IN

FIG. 42

DATA IN -- BLOCK FORMAT

CYCLE 1:	Y<B3>	Y<B2>	Y<B1>	Y<B0>
CYCLE 2:	Y<B7>	Y<B6>	Y<B5>	Y<B4>
CYCLE 3:	Y<B11>	Y<B10>	Y<B9>	Y<B8>
CYCLE 4:	Y<B15>	Y<B14>	Y<B13>	Y<B12>
CYCLE 5:	Cr<B3>	Cr<B2>	Cr<B1>	Cr<B0>
CYCLE 6:	Cr<B7>	Cr<B6>	Cr<B5>	Cr<B4>
CYCLE 7:	Cb<B3>	Cb<B2>	Cb<B1>	Cb<B0>
CYCLE 8:	Cb<B7>	Cb<B6>	Cb<B5>	Cb<B4>

1010

FIG. 43A

Y U V 4:2:2 FORMAT:

	31-24	23-16	15-8	7-0	
LITTLE-ENDIAN	Cb<B n>	Y<Bn+1>	Cr<B n>	Y<B n>	1012a
GIB-ENDIAN	Y<B n+1>	Cb<Bn>	Y<Bn>	Cr<Bn>	1012b

FIG. 43B

Y U V 4:2:0 FORMAT:

Y<Bn>	Y<B n+1>
Y<B n+2>	Y<B n+3>
Cr<B n>	Cb<B n>

FIG. 43C

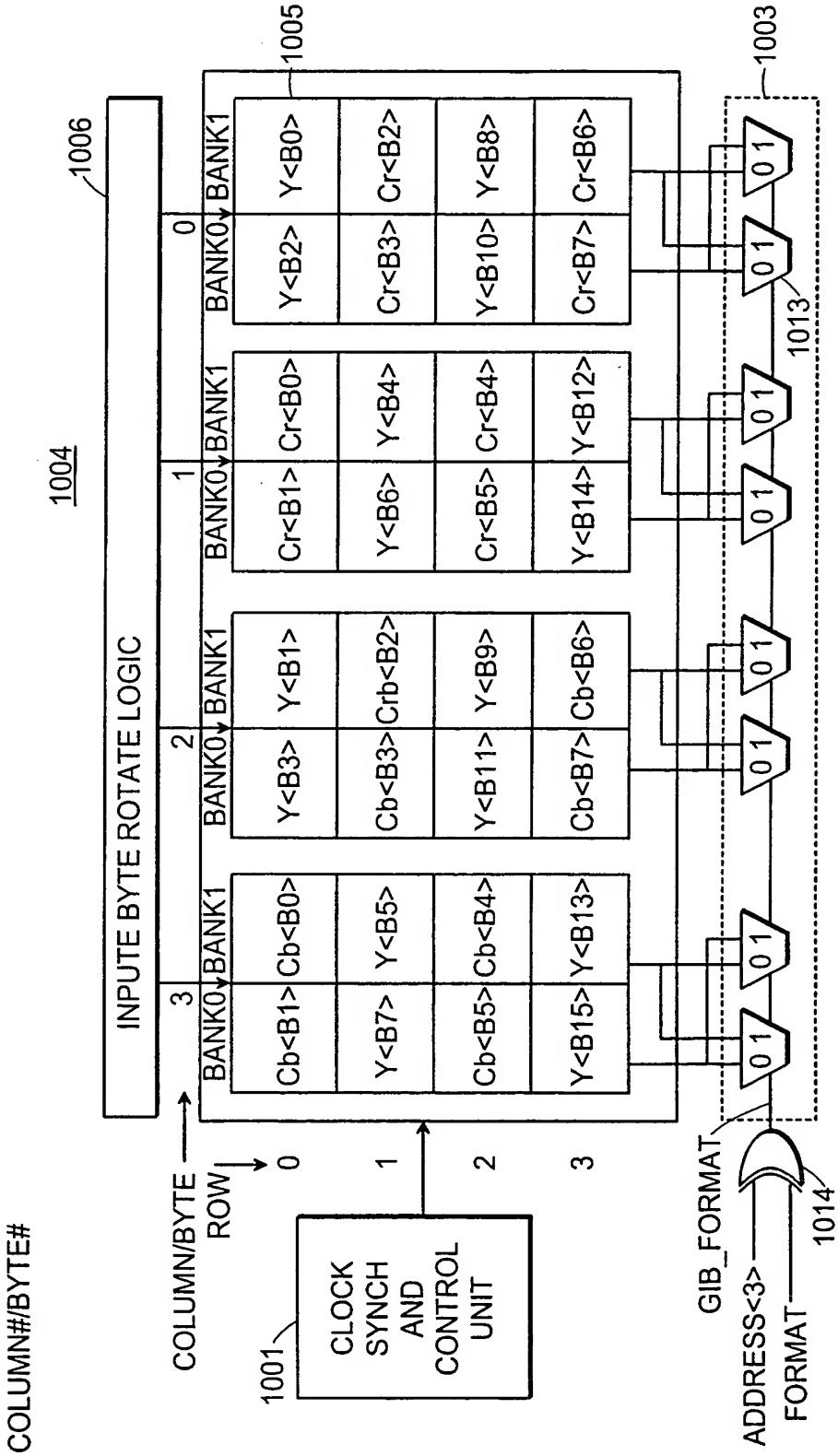

COLUMN#/BYTE#
1004
1006
INPUTE BYTE ROTATE LOGIC
1005
1003
1013
1014
1001
CLOCK SYNCH AND CONTROL UNIT
COLUMN/BYTE
ROW
GIB_FORMAT
ADDRESS<3>
FORMAT
BANK0
BANK1
Cb<B1>
Cb<B0>
Y<B7>
Y<B5>
Cb<B5>
Cb<B4>
Y<B15>
Y<B13>
Y<B3>
Y<B1>
Cb<B3>
Crb<B2>
Y<B11>
Y<B9>
Cb<B7>
Cb<B6>
Cr<B1>
Cr<B0>
Y<B6>
Y<B4>
Cr<B5>
Cr<B4>
Y<B14>
Y<B12>
Y<B2>
Y<B0>
Cr<B3>
Cr<B2>
Y<B10>
Y<B8>
Cr<B7>
Cr<B6>

FIG. 44

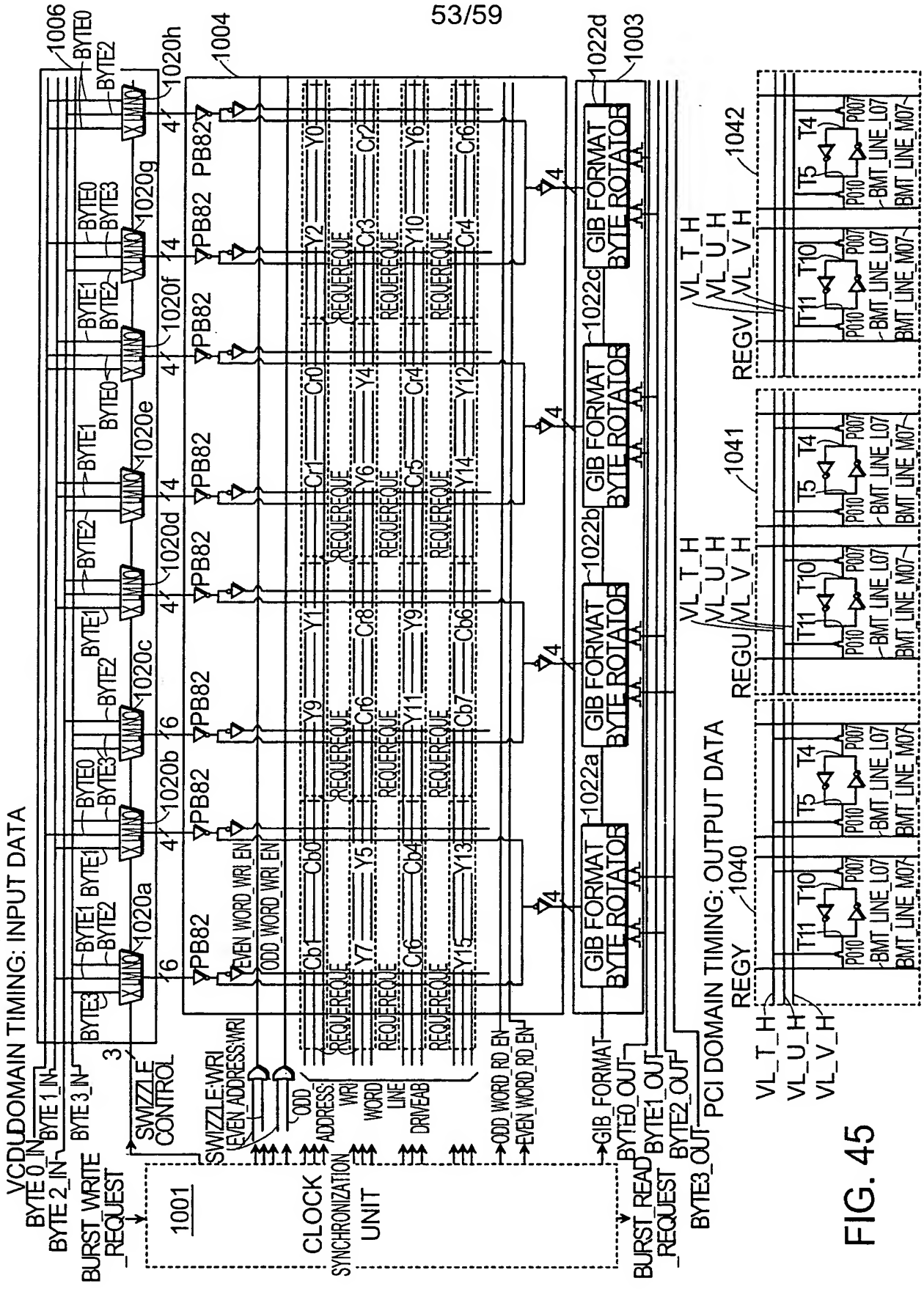

VCD DOMAIN TIMING: INPUT DATA
PCI DOMAIN TIMING: OUTPUT DATA
1001
CLOCK SYNCHRONIZATION UNIT
SWIZZLE CONTROL
BURST_WRITE_REQUEST
BURST_READ_REQUEST
BYTE0_IN
BYTE1_IN
BYTE2_IN
BYTE3_IN
BYTE0_OUT
BYTE1_OUT
BYTE2_OUT
BYTE3_OUT
SWIZZLE:WRI
EVEN_ADDRESS:WRI
ODD_WORD_WRI_EN
EVEN_WORD_WRI_EN
ODD_WORD_RD_EN
EVEN_WORD_RD_EN
GIB_FORMAT
GIB FORMAT BYTE ROTATOR
REQUEREQUE
REGU
REGV
REGY
VL_T_H
VL_U_H
VL_V_H
BMT_LINE_L07
BMT_LINE_M07
1006
1004
1003
1041
1042
1040
1022a
1022b
1022c
1022d
1020a
1020b
1020c
1020d
1020e
1020f
1020g
1020h

FIG. 45

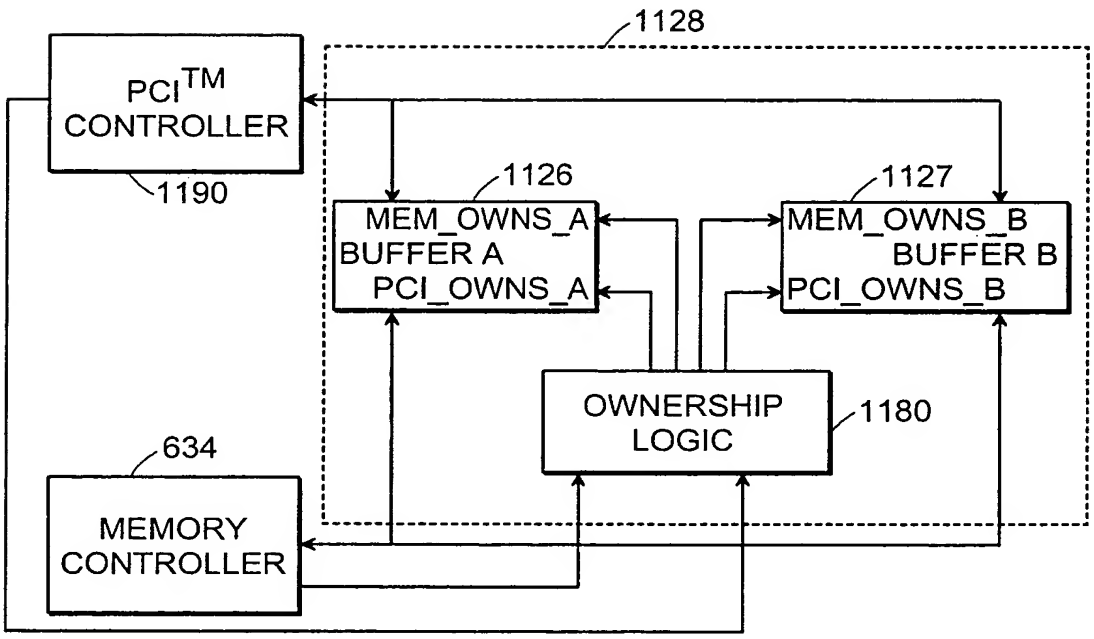

1128
PCI™ CONTROLLER
1190
1126
MEM_OWNS_A
BUFFER A
PCI_OWNS_A
1127
MEM_OWNS_B
BUFFER B
PCI_OWNS_B
OWNERSHIP LOGIC
1180
634
MEMORY CONTROLLER

FIG. 46

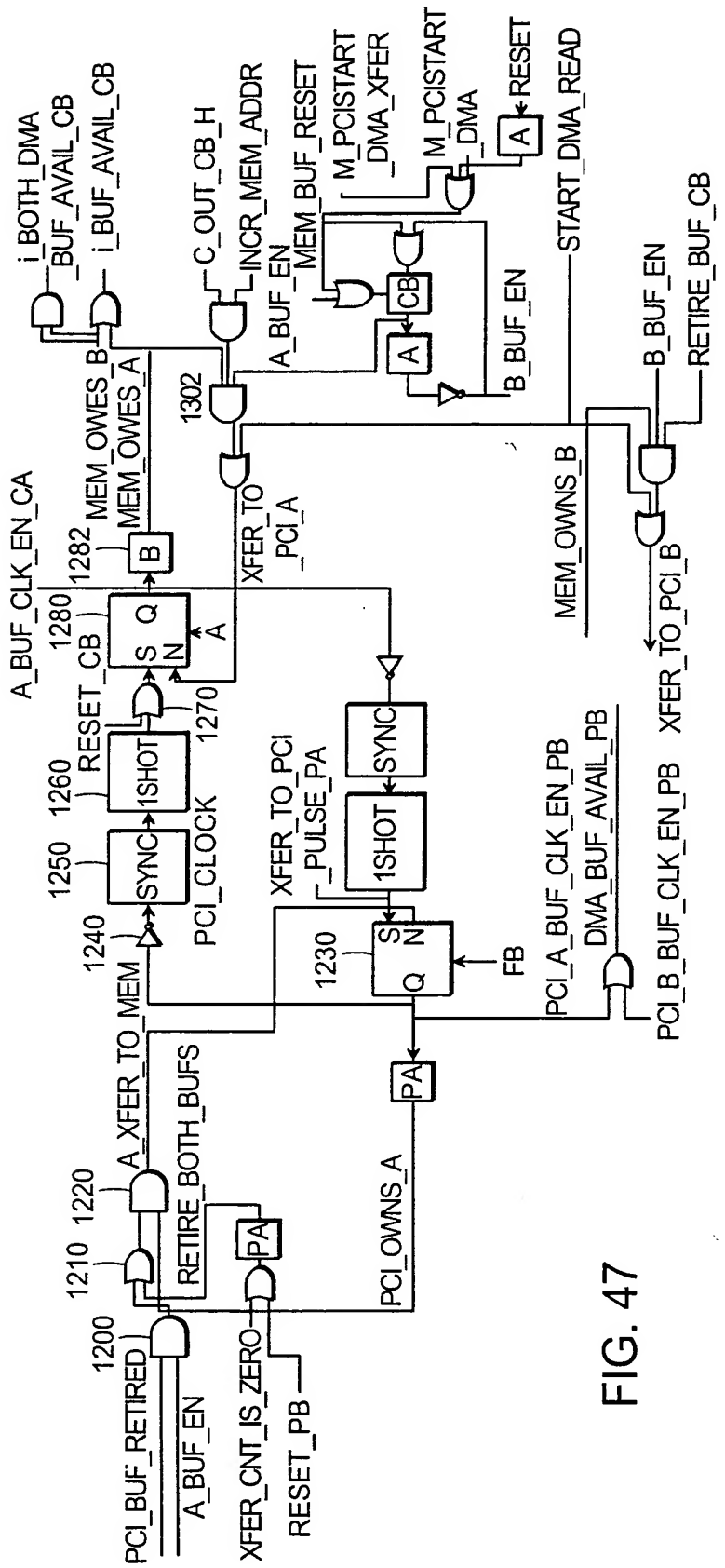

i_BOTH_DMA_BUF_AVAIL_CB
i_BUF_AVAIL_CB
C_OUT_CB_H
INCR_MEM_ADDR
A_BUF_EN
MEM_BUF_RESET
M_PCISTART_DMA_XFER
M_PCISTART_DMA
RESET
START_DMA_READ
B_BUF_EN
RETIRE_BUF_CB
MEM_OWES_B
MEM_OWES_A
1302
A_BUF_CLK_EN_CA
1282
1280
RESET_CB
1270
XFER_TO_PCI_A
MEM_OWNS_B
1260
1250
PCI_CLOCK
XFER_TO_PCI_PULSE_PA
1240
1230
FB
PCI_A_BUF_CLK_EN_PB
DMA_BUF_AVAIL_PB
PCI_B_BUF_CLK_EN_PB
XFER_TO_PCI_B
A_XFER_TO_MEM
1220
1210
RETIRE_BOTH_BUFS
PCI_OWNS_A
1200
PCI_BUF_RETIRED
A_BUF_EN
XFER_CNT_IS_ZERO
RESET_PB
SYNC
1SHOT
S
Q
N
A
B
CB
PA

FIG. 47

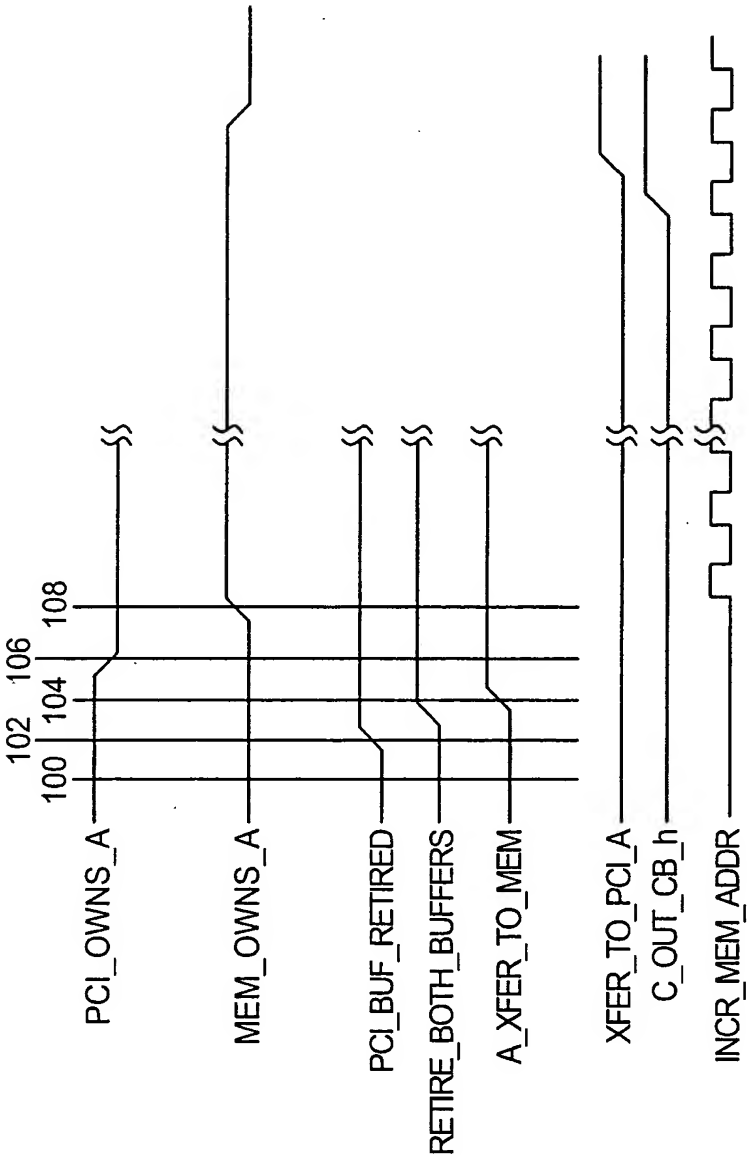

100
102
104
106
108
PCI_OWNS_A
MEM_OWNS_A
PCI_BUF_RETIRED
RETIRE_BOTH_BUFFERS
A_XFER_TO_MEM
XFER_TO_PCI_A
C_OUT_CB_h
INCR_MEM_ADDR

FIG. 48

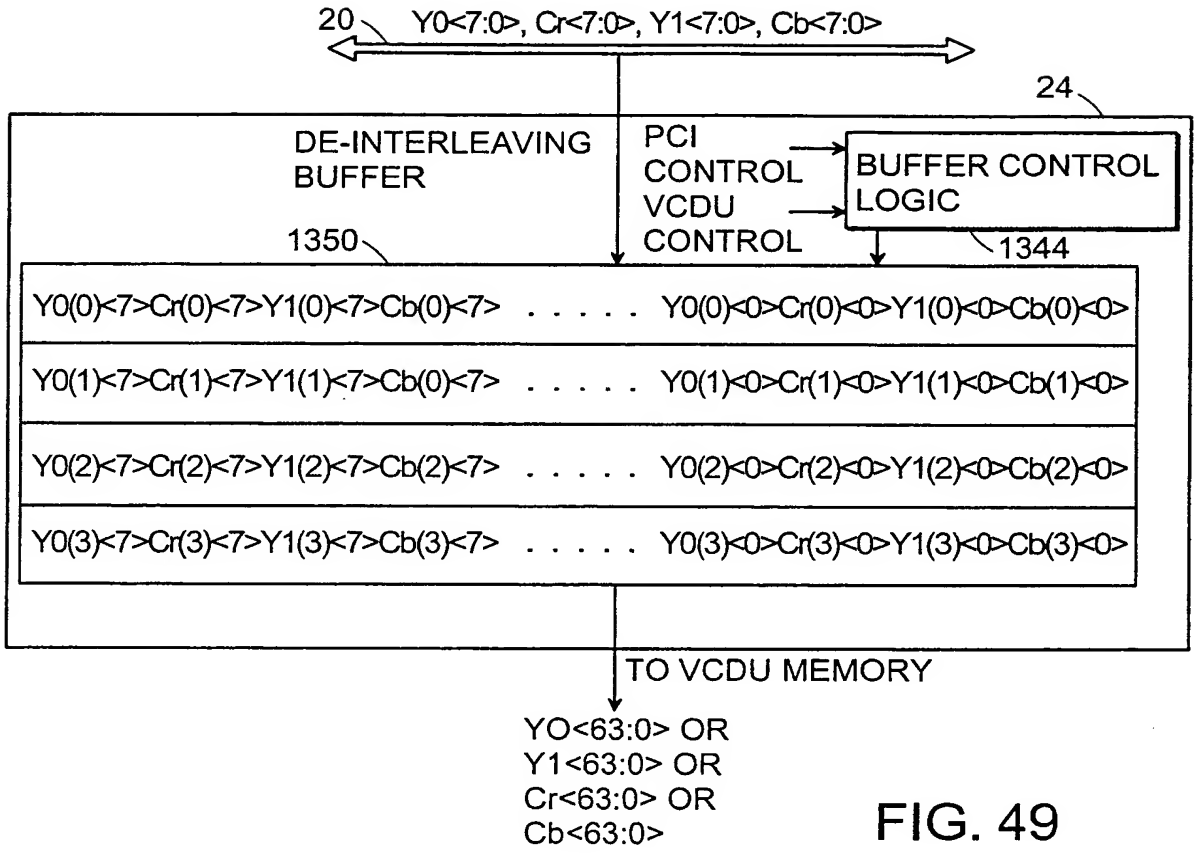

20
Y0<7:0>, Cr<7:0>, Y1<7:0>, Cb<7:0>
24
DE-INTERLEAVING BUFFER
PCI CONTROL
VCDU CONTROL
BUFFER CONTROL LOGIC
1350
1344
Y0(0)<7>Cr(0)<7>Y1(0)<7>Cb(0)<7> . . . . . Y0(0)<0>Cr(0)<0>Y1(0)<0>Cb(0)<0>
Y0(1)<7>Cr(1)<7>Y1(1)<7>Cb(0)<7> . . . . . Y0(1)<0>Cr(1)<0>Y1(1)<0>Cb(1)<0>
Y0(2)<7>Cr(2)<7>Y1(2)<7>Cb(2)<7> . . . . . Y0(2)<0>Cr(2)<0>Y1(2)<0>Cb(2)<0>
Y0(3)<7>Cr(3)<7>Y1(3)<7>Cb(3)<7> . . . . . Y0(3)<0>Cr(3)<0>Y1(3)<0>Cb(3)<0>
TO VCDU MEMORY
YO<63:0> OR
Y1<63:0> OR
Cr<63:0> OR
Cb<63:0>

FIG. 49

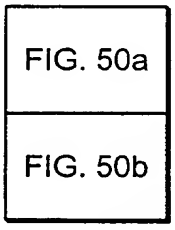
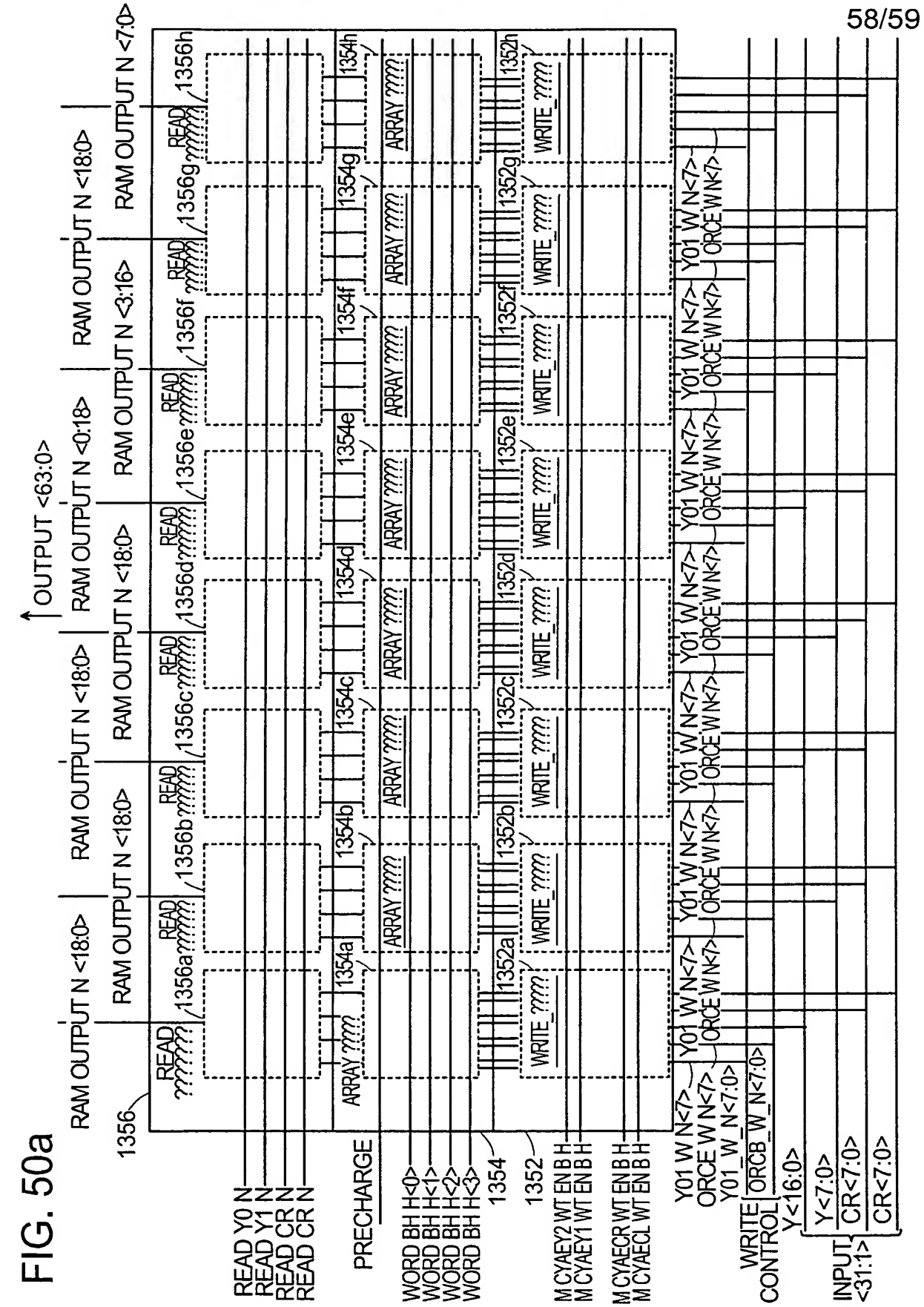


FIG. 50a
FIG. 50b

FIG. 50

FIG. 50a


OUTPUT <63:0>
RAM OUTPUT N <18:0>
RAM OUTPUT N <3:16>
RAM OUTPUT N <7:0>
1356
1356a
1356b
1356c
1356d
1356e
1356f
1356g
1356h
READ ??????
ARRAY ?????
WRITE_?????
1354a
1354b
1354c
1354d
1354e
1354f
1354g
1354h
1352a
1352b
1352c
1352d
1352e
1352f
1352g
1352h
1354
1352
READ Y0 N
READ Y1 N
READ CR N
READ CR N
PRECHARGE
WORD BH H<0>
WORD BH H<1>
WORD BH H<2>
WORD BH H<3>
M CYAEY2 WT EN B H
M CYAEY1 WT EN B H
M CYAECR WT EN B H
M CYAECL WT EN B H
Y01 W N<7>
ORCE W N<7>
Y01_W_N<7:0>
WRITE CONTROL
ORCB_W_N<7:0>
Y<16:0>
INPUT <31:1>
Y<7:0>
CR<7:0>
CR<7:0>

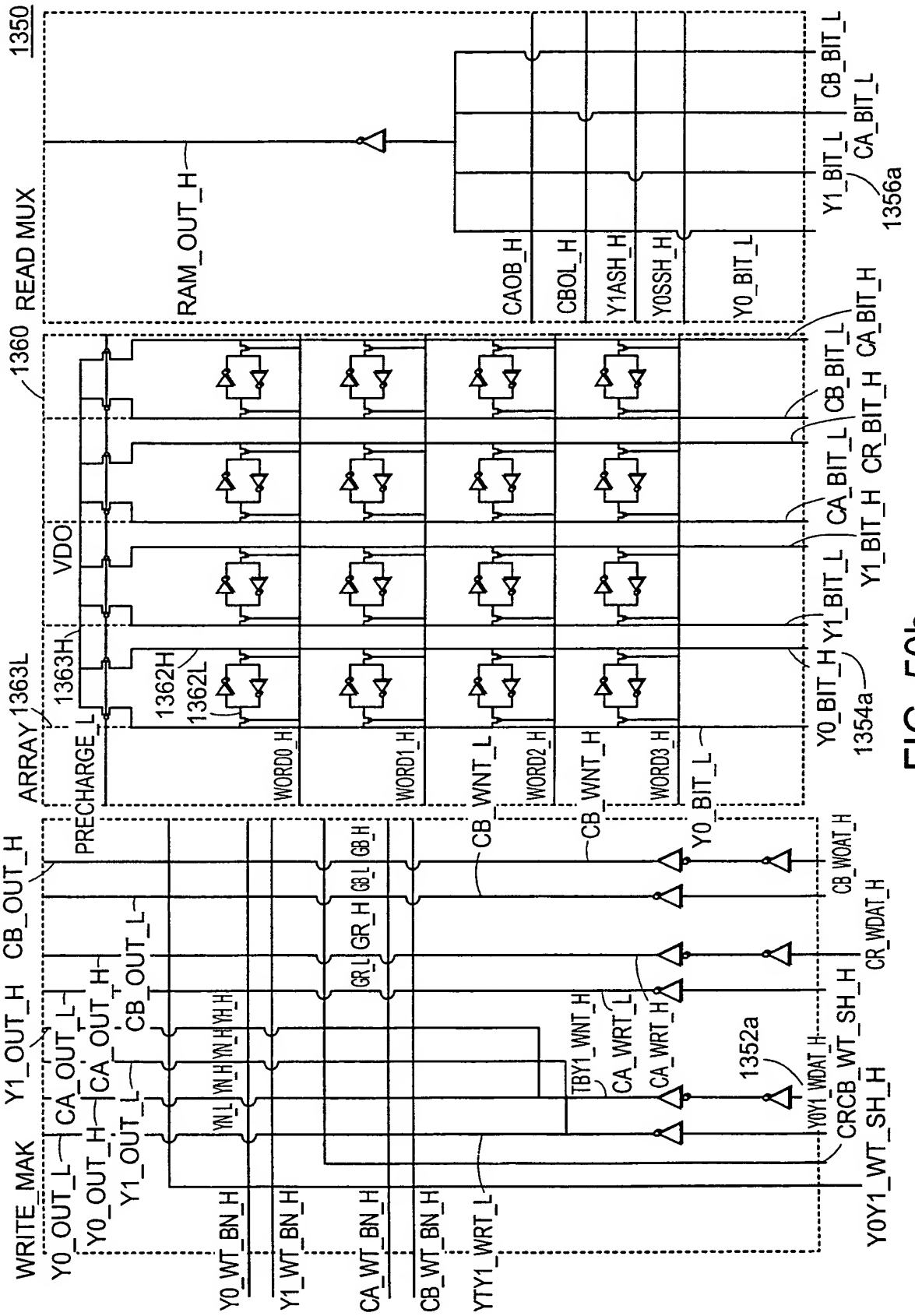

1350
READ MUX
RAM_OUT_H
CAOB_H
CBOL_H
Y1ASH_H
Y0SSH_H
Y0_BIT_L
Y1_BIT_L
CB_BIT_L
CA_BIT_L
1356a
1360
ARRAY
VDO
PRECHARGE_L
1363H
1363L
1362H
1362L
WORD0_H
WORD1_H
WORD2_H
WORD3_H
CB_WNT_L
CB_WNT_H
Y0_BIT_H
Y1_BIT_H
CR_BIT_H
1354a
WRITE_MAK
Y0_OUT_L
Y0_OUT_H
Y1_OUT_L
Y1_OUT_H
CA_OUT_L
CA_OUT_H
CB_OUT_L
CB_OUT_H
YN_L
YN_H
YH_L
YH_H
GR_L
GR_H
GB_L
GB_H
Y0_WT_BN_H
Y1_WT_BN_H
CA_WT_BN_H
CB_WT_BN_H
YTY1_WRT_L
TBY1_WNT_H
CA_WRT_L
CA_WRT_H
1352a
Y0Y1_WDAT_H
CRCB_WT_SH_H
Y0Y1_WT_SH_H
CB_WDAT_H
CR_WDAT_H

FIG. 50b